

Projecte de Fi de Carrera  
**Enginyer Industrial**

**Disseny d'un sistema robòtic  
d'escaneig, marcat i polit de superfícies  
en un entorn industrial**

**MEMÒRIA  
ANNEX**

**Autor:** Maurici Pàmias Prohias  
**Directora:** Alícia Casals Gelpí  
**Co-director:** Xavier Giralt Ludevid  
**Convocatòria:** Novembre 2014



Escola Tècnica Superior  
d'Enginyeria Industrial de Barcelona



## Resum

L'objectiu del projecte és el desenvolupament d'un sistema autònom d'escaneig, polit i marcat de superfícies, dins el marc del procés industrial de l'acabat en la fabricació de peces de grans dimensions. Per al desenvolupament del projecte es disposa d'un entorn a petita escala amb un braç robòtic, un escàner puntual i una eina de polit.

El projecte està constituït per tres blocs. El primer consisteix en el desenvolupament i programació d'un software per a la generació automàtica de les trajectòries del robot, per als processos d'escaneig, de marcat i de polit. El segon és el disseny de l'element terminal que subjecta les eines dels processos, usant un software CAD i la seva posterior fabricació. I finalment, el disseny i implementació dels mòduls electrònics necessaris.

En la part de l'escaneig l'objectiu és la generació de models en 3 dimensions de superfícies d'objectes. El procediment ha de ser capaç d'adaptar-se a la geometria i dimensió de l'objecte escanejat, arribant a desenvolupar tres mètodes diferents de generació de trajectòries del robot.

Amb el polit es vol obtenir una rèplica de l'objecte que s'ha escanejat. A partir de les dades obtingudes amb el procés d'escaneig, el robot ha de ser capaç d'eliminar el material sobrant d'un motlle fins que quedi una reproducció de l'objecte.

En el marcat la finalitat és que el robot dibuixi sobre una superfície uns traços indicats prèviament.

Per poder utilitzar totes aquestes eines s'ha hagut de dissenyar un mètode calibració per cada una d'elles. Aquest apartat del projecte s'ha desenvolupat en gran part amb MATLAB.

Aquest és un projecte d'enginyeria multi-disciplinar, en el que s'han combinat coneixements de programació, robòtica, electrònica i disseny mecànic per tal d'obtenir uns resultats que compleixin amb suficiència els objectius proposats.



# Índex

Resum .....	1
Prefaci .....	5
Introducció.....	7
Capítol 1	
Arquitectura.....	9
1.1. Entorn de programació.....	10
1.2. Robot .....	12
1.2.1. Característiques del robot.....	12
1.2.2. Arquitectura de comunicació amb la controladora del robot .....	13
1.2.3. Control de la trajectòria robot .....	14
1.3. Placa microcontroladora .....	19
1.3.1. Hardware de la placa microcontroladora .....	20
1.3.2. Comunicació entre l'aplicació i la placa microcontroladora .....	22
Capítol 2	
Funcions del sistema desenvolupat .....	23
2.1. Funció escaneig .....	23
2.1.1. Components de l'escaneig .....	24
2.1.2. Funcionament del procés d'escaneig.....	24
2.1.3. Electrònica del procés d'escaneig .....	33
2.2. Funció dibuixar .....	36
2.2.1. Mecanisme del marcador.....	36
2.2.2. Funcionament del procés marcat .....	39
2.2.3. Electrònica del marcat.....	42
2.3. Funció de polit .....	44
2.3.1. Components .....	44
2.3.2. Funcionament del procés de polit .....	45
2.4. Funció apuntar.....	52

## Capítol 3

Disseny i fabricació del capçal ..... 55

3.1. Platina ..... 56

3.2. Capçal de plàstic ..... 56

## Capítol 4

Calibració ..... 61

## Capítol 5

Pressupost i consideracions ambientals..... 65

5.1. Pressupost ..... 65

5.2. Consideracions ambientals..... 68

Conclusions ..... 69

Agraïments..... 71

Bibliografia..... 73

Annex..... 75

## Prefaci

El present projecte s'ha realitzat al laboratori de robòtica del Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial (ESAI), de la UPC. En aquest departament es desenvolupen varis projectes, un dels quals és el projecte EAPS (Estación Automática de Pulido de grandes Superficies), del qual aquest treball en forma part. El projecte consisteix l'automatització dels processos d'escanejat i polit de grans superfícies.

Per la realització del projecte s'ha comptat amb un robot ABB IRB140, un ordinador, eines d'electrònica, un escàner, un sistema multi-eina per polir, i material fungible per a la implementació de l'equip de proves. Tot i que gran part de l'equip utilitzat en aquest projecte, com el robot i l'escàner, tenen característiques diferents als usats en el projecte EAPS, ha estat un punt de partida per desenvolupar el projecte en bones condicions, i una garantia de que el resultat és extrapolable al robot de grans dimensions.

L'automatització de processos ofereix molts avantatges en els processos industrials, i des de fa temps està substituint els processos tradicionals. Entre les millores que ofereix es poden destacar un treball més ràpid, de més qualitat i amb menys costos.

La motivació d'aquest projecte és aconseguir automatitzar un procés amb èxit i conèixer i dominar un camp que ofereix moltes possibilitats i que es pot exportar a molts àmbits de l'enginyeria. També és un factor d'interès el fet de que en aquest projecte s'hi barregin diferents aspectes de l'enginyeria com la programació, l'electrònica i el disseny en CAD.



## Introducció

Aquest projecte està englobat dins un projecte de dimensions més grans anomenat EAPS (Estación Automática de Pulido de grandes Superficies). El projecte EAPS té com a objectiu l'automatització del procés de polit de grans superfícies per a la producció de vagons de tren. Aquest procés, realitzat manualment, requereix un operari que detecti les imperfeccions en les superfícies, posi massilla a les zones que falti material, i poleixi en les que en sobra.

El projecte EAPS pretén substituir l'operari en les fases de l'escanejat i el polit per un robot que les realitzi autònomament. El procés automatitzat segueix l'estructura següent:

1. El robot escaneja la superfície amb un escàner lineal, obtenint-ne un model en 3D. Per a fer això el robot ha de seguir una trajectòria tal que l'escàner cobreixi tota la superfície.
2. Es compara la imatge obtinguda amb un model ideal de la superfície, trobant les zones on falta material.
3. Aquestes zones han de ser omplertes amb massilla. Aquesta és la única tasca que no s'ha automatitzat, i que ha de ser realitzada per l'operari. El robot indica a l'operari les zones de la superfície on ha d'aplicar la massilla amb un sistema de marcat.
4. Un cop aplicada i assecada la massilla el robot torna a escanejar per, posteriorment, tornar a comparar la imatge amb el model ideal de la superfície i polir en les zones on detecta material sobrant.

El projecte que s'exposa a continuació té com a objectiu desenvolupar la part del software de generació de les trajectòries del robot per les tasques d'escaneig, marcat i polit.

Per a realitzar les proves s'ha comptat amb un robot ABB model IRB 140, un robot de menors dimensions que l'usat en el projecte EAPS, però amb el mateix esquema de funcionament. A l'element terminal del robot s'hi han acoblat les eines necessàries per la realització de cada tasca (Figura I.1).

El projecte està constituït per tres blocs. El primer consisteix en el desenvolupament i programació d'un software per a la generació automàtica de les trajectòries del robot, per als processos d'escaneig, de marcat i de polit. El segon és el disseny del capçal que



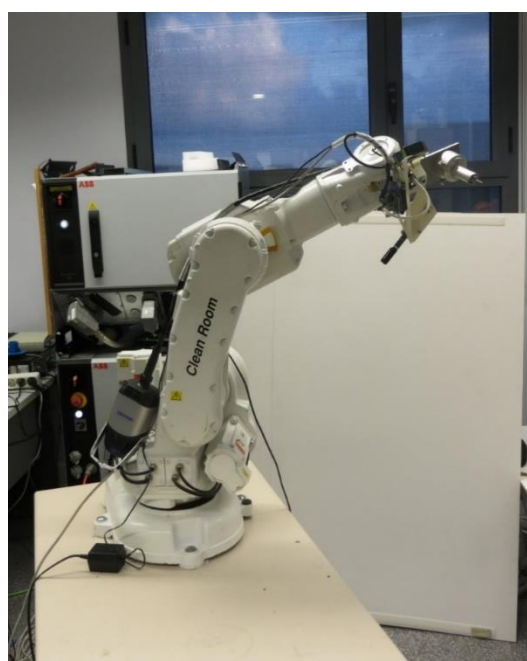
subjecta les eines dels processos, i la seva posterior fabricació. I finalment, el disseny, programació i implementació dels mòduls electrònics necessaris.

El procés d'escaneig s'ha desenvolupat per tal que el robot pugui escanejar superfícies d'objectes de dimensions i geometries variades. En el projecte EAPS s'utilitza un escàner lineal, en canvi per aquest projecte s'ha disposat d'un escàner puntual. El software s'ha desenvolupat per a aquest últim, procurant que l'adaptació del software a l'escàner lineal sigui senzilla.

La tasca de marcat el robot utilitza un marcador per fer traços sobre una superfície. En el projecte EAPS, els traços defineixen el contorn d'àrees tancades que indiquen a l'operari les zones on s'ha d'aplicar la massilla.

La tasca de polit consisteix en l'eliminació de material sobrant d'una superfície per obtenir el resultat desitjat. S'ha desenvolupat el software que genera la trajectòria del robot per a aquest procés, i s'ha acoblat un sistema multi-eina al capçal del robot que permet utilitzar diferents eines per polir: broques, discs, raspalls, etc. Per provar la validesa de l'aplicació desenvolupada s'ha marcat l'objectiu d'obtenir una rèplica d'un objecte eliminant material d'un motlle. La forma de l'objecte s'obté escanejant l'objecte amb l'aplicació anteriorment esmentada. Anàlogament al projecte EAPS, l'objecte escanejat representa el model ideal de la superfície, i el motlle la superfície abans de ser polida.

Les aplicacions s'han desenvolupat amb Microsoft Visual C++. Pel disseny del capçal s'ha fet us del SolidWorks i el disseny dels circuits electrònics s'ha fet amb EAGLE. Pel tractament de dades s'ha fet us del MATLAB i l'Excel.



**Figura I.1:** Fotografia del robot utilitzat en les proves amb les eines acoblades

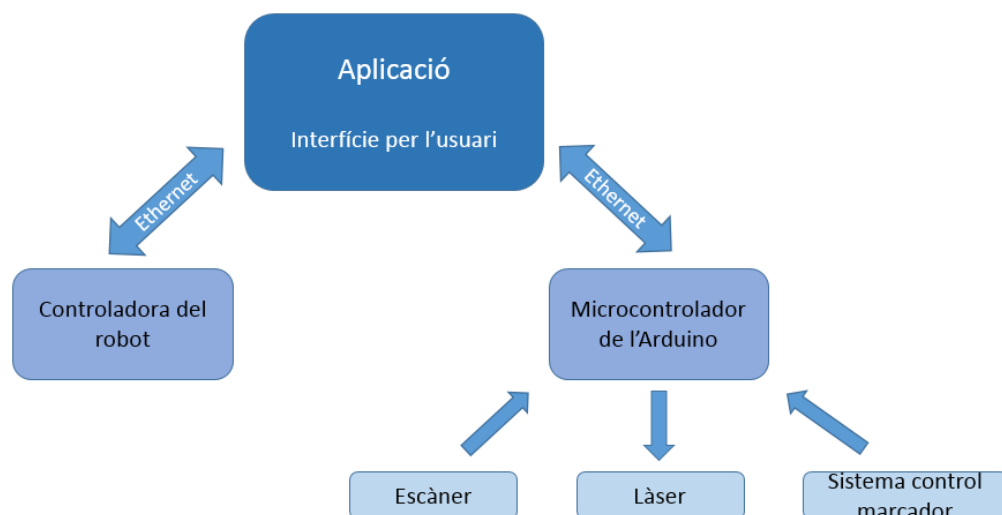
# Capítol 1

## Arquitectura

Aquest projecte pretén que els procediments a seguir siguin el més automàtics possible. Tot i això és necessari un usuari que intervingui en alguns punts del procés. Per això s'ha volgut desenvolupar unes interfícies gràfiques d'usuari (GUI) clares, senzilles i intuïtives, perquè l'usuari pugui executar totes les aplicacions havent rebut una explicació bàsica del seu funcionament.

Alguns procediments, com la calibració d'eines i el tractament de les dades obtingudes amb l'escàner, que es fan amb MATLAB i Excel respectivament, no estan automatitzats, i requereixen un nivell de coneixement alt del programari i del projecte. Es deixa com a objectiu futur l'automatització d'aquests processos mitjançant el desenvolupament del software equivalent en aplicacions C++.

L'usuari controla tot el procés des d'un computador on s'executen les aplicacions. Tal i com es descriu a la Figura 1.1, l'aplicació del sistema s'ha de comunicar amb dos altres processadors: la controladora del robot i la placa microcontroladora, que, col·locada a l'element terminal del robot, gestiona els diferents mòduls electrònics. La comunicació entre aquests tres computadores es fa mitjançant un protocol de comunicació client/servidor basat en TCP/IP sobre Ethernet.



**Figura 1.1:** Esquema de la comunicació entre dispositius

## 1.1. Entorn de programació

Les aplicacions s'han desenvolupat en C++, usant coma entorn de programació el Microsoft Visual Studio. El Visual Studio facilita el desenvolupament d'aplicacions que es comuniquin entre diferents estacions de treball. Els coneixements previs i l'experiència de treballs anteriors en el mateix llenguatge i entorn de programació han motivat l'elecció d'aquest entorn.

En el Visual Studio s'han desenvolupat tres funcions principals, les que controlen els processos d'escaneig, de polit i de marcat. Totes les aplicacions tenen estructures semblants, i compten amb diferents blocs de funcions encarregats de tasques diferents. Aquestes tasques són:

1. **Inicialització de l'aplicació:** En aquest bloc s'inicialitzen les variables necessàries i es llegeixen els fitxers de dades, si és que n'hi ha.
2. **Generació de trajectòries:** En totes les tasques, menys la del làser, hi ha un bloc de funcions que s'encarreguen de generar, parcial o totalment, les trajectòries que seguirà el robot durant el procés.
3. **Comunicació amb robot i placa microcontroladora:** Aquestes funcions tenen la funció d'establir la connexió entre les diferents estacions de treball. El seu funcionament s'explica amb detall més endavant, en l'apartat 1.2.
4. **Càlculs, generació de missatges i recopilació de dades:** Són les funcions que s'executen mentre el robot està realitzant la tasca. En elles es fan els càlculs necessaris per generar els missatges que s'envien a la controladora del robot i a la placa microcontroladora i, en el cas de l'escaneig, desa la informació que prové de l'escàner.

La placa microcontroladora usada és la Arduino Ethernet Rev. 3. S'ha escollit aquest model ja que té una estructura ideal per la implantació de tots els mòduls i components electrònics necessaris i permet la connexió amb el computador que executa les aplicacions via Ethernet. La seva programació també s'ha realitzat en llenguatge C++. L'entorn de programació és gratuït i disponible a la pàgina web oficial d'Arduino. És un entorn específic per a la programació d'aquestes plaques microcontroladores. Un cop compilat el programa, permet carregar-lo immediatament al microcontrolador, així com usar un monitor del port sèrie per a seguir i depurar el programa.

En moments puntuals del projecte ha calgut usar eines software addicionals. La part de calibració de les eines ha requerit l'ús del MATLAB, mentre que la part de tractament de dades s'ha realitzat amb Excel.

## 1.2. Robot

El robot utilitzat ha estat el model IRB 140 de la marca ABB.



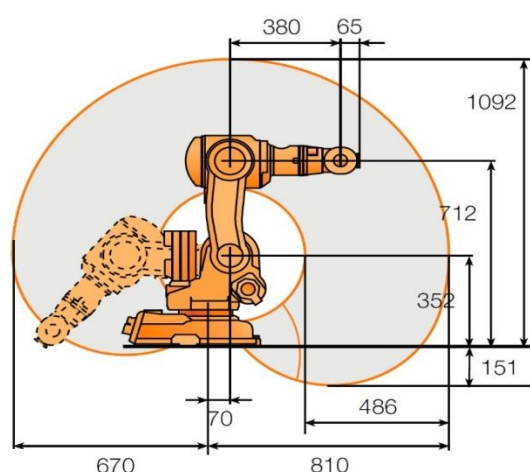
**Figura 1.2:** Robot ABB model IRB140

### 1.2.1. Característiques del robot

L'IRB 140 és un robot industrial de sis eixos capaç de suportar càrregues de fins a 5 kg. És un robot compacte i el rang de treball és limitat, essent 810 mm la distància màxima a la que es pot allunyar l'element terminal de la base del robot en el pla horitzontal.

Eix	Rang de treball
1	360°
2	200°
3	280°
4	Il·limitat ( 400° per defecte)
5	240°
6	Il·limitat (800° per defecte)

**Taula 1.1:** Rang de treball de les sis articulacions del robot



**Figura 1.3:** Rang de treball del robot

### 1.2.2. comunicació amb la controladora del robot

Com s'ha comentat en la introducció d'aquest capítol, les aplicacions s'han de comunicar constantment amb la controladora del robot, per enviar-li les ordres de moviment i rebre informació de la posició actual del robot en qualsevol instant.

L'estructura de programació que permet la comunicació és comuna a totes les aplicacions, i consta de quatre funcions.

- `void OnTimer ()`

En començar a executar-se el programa, s'activa un temporitzador per cada canal de comunicació que es vol establir via TCP/IP sobre Ethernet (controladora del robot, Arduino o d'altres). Aquest temporitzador fa que s'accedeixi periòdicament a la funció `OnTimer`, a on, a través de la llibreria `OurSocket`, s'intenta començar una comunicació amb el dispositiu que tingui la IP i el número de port indicats.

- `void OnConnected ()`

Aquesta funció s'activa quan es rep un missatge que prové del servidor al qual ens volem connectar, especificant que la connexió s'ha establert. Dins de la funció, es destrueix el temporitzador del client en qüestió i s'habiliten totes les funcions que requereixen d'aquesta connexió.

- `void OnClosed ()`

El punter entra en aquesta funció quan rep un missatge d'un dels servidor comunicant que la connexió s'ha acabat. La seva comesa és inhabilitar totes les funcions vinculades a aquesta connexió i tornar a crear el temporitzador perquè el servidor es pugui tornar a connectar al client tan bon punt aquest estigui disponible.

- `void OnRecieve ()`

La funció *OnRecieve* s'activa cada cop que es rep un missatge d'un dels client. És essencial per a la comunicació continuada entre servidor i client, ja que s'encarrega de rebre i interpretar els missatges que arriben cada cop que el robot es mou o que es demana una dada a la placa controladora i de, a partir d'aquesta informació, determinar quin és el següent pas que l'aplicació ha de seguir.

### 1.2.3. Control de la trajectòria robot

La programació del moviment del robot té una estructura molt semblant en totes les aplicacions del projecte. En aquest apartat s'expliquen alguns conceptes comuns en totes elles. El primer és el **software de la controladora del robot**, que és amb el que es comuniquen les aplicacions i el que controla en última instància els moviments del robot. El segon és el **control dels moviments**, que es pot indicar a nivell de gir de les articulacions o en coordenades cartesianes. Els moviments també es poden donar en **desplaçament incremental o en absolut**. També s'explica quin mètode es fa servir per **enllaçar moviments** evitant parades i arrancades de motor, i a definir la trajectòria del robot afegint **punts de parada intermitjos**. Per últim s'expliquen els càlculs necessaris per **obtenir la matriu de desplaçament** que s'ha d'indicar al robot per situar l'eina que s'està utilitzant al lloc desitjat.

#### Software de la controladora del robot

La comunicació de l'aplicació de control amb la controladora del robot i placa electrònica es realitza via Ethernet gràcies a la llibreria `OurSocket.lib`. Aquesta llibreria permet a un computador establir comunicació amb altres, indicant la IP i el port de cada una d'elles. En aquest projecte, la computadora que executa els programes s'ha de comunicar amb la controladora del robot i la placa.

Per facilitar l'enviament i recepció d'ordres entre les aplicacions i el programa intern del robot, s'ha utilitzat un conjunt de funcions que fan d'interpret. Aquestes funcions transformen les ordres que es volen donar al robot en un missatge que la controladora pugui entendre, per, posteriorment, enviar-lo a la controladora del robot.

Un cop el robot ha realitzat el moviment indicat, la controladora retorna un missatge en el qual indica la posició actual i altra informació addicional. Aquest missatge es fa servir com a indicatiu de que el robot ha acabat un moviment i ja es pot enviar la següent comanda.

Tots els missatges que s'envien al robot contenen un codi, que és retornat pel robot en el missatge de retorn. Aquest codi serveix perquè el programa sàpiga quina funció ha d'executar a continuació de rebre el missatge.

### Control del moviment a nivell de gir de les articulacions o coordenades cartesianes

La controladora del robot pot rebre les ordres de moviment de diferents maneres, ja sigui indicant els graus de gir de cada una de les sis articulacions, o indicant la posició X, Y, Z i l'orientació de l'element terminal respecte la base del robot. En aquest últim cas, és la mateixa controladora del robot que fa la cinemàtica inversa i calcula els graus de gir de les articulacions, ja que, en última instància, és aquesta la informació que s'utilitza per moure el robot. En el cas d'indicar la posició en coordenades cartesianes s'ha d'especificar també l'orientació. Per això hi ha tres opcions diferents: enviar la matriu de rotació, els angles d'Euler o el quaternió.

A l'enviar un comandament en coordenades cartesianes el robot pot escollir anar-hi generant diferents configuracions. Aquesta ambigüitat fa que sigui més segur enviar comandaments de tipus articulacions. Tot i això, els càlculs de posicions s'han fet sempre amb les coordenades cartesianes i les matrius de rotació, ja que resulta molt complicat fer-ho amb els graus de gir de les articulacions. És per això que ha calgut una funció per traduir la posició de coordenades cartesianes a coordenades articulars:

- `void` CartesianesJoints ()

Aquesta funció té com a *input* el vector de posició i la matriu de rotació. La sortida és un vector de 6 dimensions que indica els graus de gir de les 6 articulacions. La funció permet controlar quina configuració s'utilitza per arribar a la posició.

### Desplaçament incremental o absolut

Tant si es donen les ordres en coordenades articulars o en coordenades cartesianes, es pot fer de manera incremental o absoluta. Quan es fa de forma incremental, la nova posició s'indica respecte la posició anterior. En canvi, en el cas de moviment absolut, la nova posició té com a referència la base del robot. Un exemple per aclarir-ho:

Si el robot es troba en el punt de l'espai (30,10,40) i se li aplica un vector de desplaçament incremental (0,100,0), la posició final serà (30,110,40). Si el que s'aplica és un moviment absolut a la posició (0,100,0), aquesta serà la posició final.

Excepte en alguns casos comptats, en el projecte s'ha utilitzat el posicionat absolut. S'ha escollit aquesta opció ja que permet saber en tot moment la posició actual del robot, i evita la possibilitat que hi ha en l'incremental de l'error acumulat.

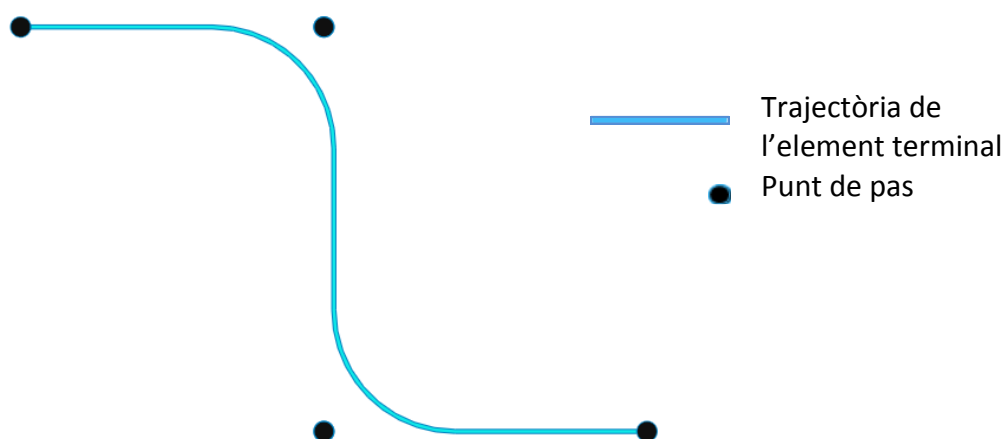


## Enllaçament de moviments

La controladora interna del robot permet la possibilitat d'enllaçar moviments per evitar que hi hagi parada i arrancada cada cop que el robot arriba a un punt en una seqüència de moviments. Aquesta opció és molt adequada en processos en que la trajectòria del robot està definida per una seqüència de punts molt pròxims. Tot i així no és vàlida quant es vol controlar la posició del robot i els seus moviments amb molta exactitud. Això és degut a dos factors que s'expliquen a continuació.

Un d'ells és que la controladora del robot envia el missatge de retorn abans d'haver acabat el moviment, ja que, d'aquesta manera, se li pot enviar el missatge que conté la posició del següent punt amb prou temps perquè pugui adaptar la trajectòria. Això implica que la posició que indica el missatge que es rep del robot no correspon a la que se li havia indicat, ja que en el moment de calcular la posició actual el robot s'està desplaçant.

L'altre factor fa referència als canvis que ha d'aplicar la controladora interna en la trajectòria per tal d'evitar canvis bruscos de direcció. Per exemple, en els cas que no s'apliqui la funció, un canvi de direcció de 90° obligaria el robot a frenar per complet i a tornar a arrancar. La funció zona, canvia l'angle recte de la trajectòria per una corba d'un cert radi. Com es veu en la figura 1.4, aquesta modificació de la trajectòria fa que el robot no passi exactament per algun dels punts indicats.



**Figura 1.4:** Trajectòria resultant a l'aplicar la funció zona

El paràmetre *zona* és un enter que pot prendre els valors 0, 1, 2, 5, 10 i 20. Amb valor 0, la funció està desactivada. A partir del valor 1 la funció ja està activada, i com més alt és el valor de la variable, més aviat es fa el canvi de trajectòria.

En l'interpret hi ha una funció que permet donar un valor a aquesta variable. Per millorar alguns procediments s'ha creat una funció que permet canviar el valor de la variable i donar una posició al mateix temps.

- `CString absolute_joints_zone (joint1, joint2, joint3, joint4, joint5, joint6, zone, code)`

S'ha creat també una funció que a més permet canviar la velocitat del robot. Aquesta funció és necessària per processos com el polit, en el que el robot es mou a diferents velocitats en funció de si està polint o s'està desplaçant.

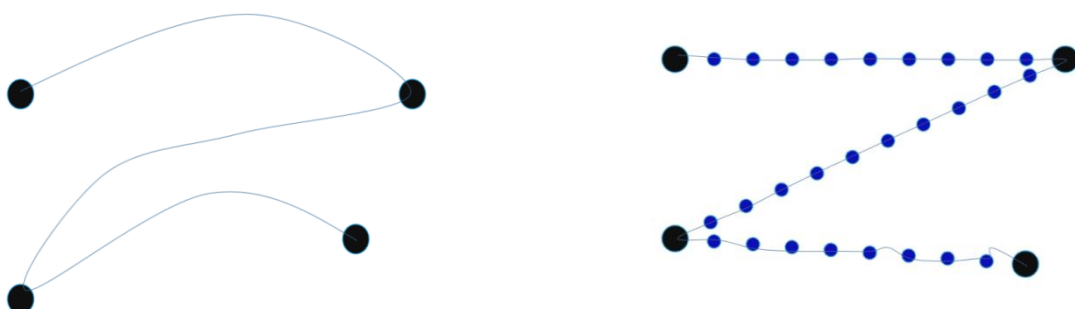
- `CString absolute_joints_zone_vel (joint1, joint2, joint3, joint4, joint5, joint6, zone, vel, code)`

### Definició de la trajectòria del robot amb punt intermitjos

A l'hora de moure el robot pot ser complicat predir la trajectòria que seguirà entre una posició i l'altra. En alguns casos aquesta trajectòria no té importància, però en molts altres sí, ja sigui per evitar que el robot passi per zones on pugui col·lisionar amb algun objecte, o perquè ha de fer una funció durant aquest recorregut, com és el cas del polit.

Per aquests casos s'ha fet servir un mateix recurs en tots els processos, consistent en la divisió d'aquests desplaçaments en moviments més curts. Així s'obliga al robot a passar pels punts entremitjos entre el punt d'origen i el de destí. La generació d'aquests punts es pot observar a la Figura 1.6.

Per aquest tipus de moviment és molt útil la funció explicada en l'apartat anterior, ja que pot resultar molt molest que en un trajecte dividit en, per exemple, dos-cents sub-trajectes, el robot s'hagi d'aturar i tornar a arrancar en cada un d'ells.



**Figura 1.5:** Trajectòria del robot abans i després de la generació de punts intermitjos

### Tractament de matrius

En molts casos, el moviment que es vol determinar no és el del monyó del robot, sinó el de l'element terminal que s'està utilitzant. Tot i així la matriu de translació i rotació que s'envia al robot és la que correspon al monyó respecte a la base d'aquest. Per a obtenir aquesta matriu, s'han de realitzar un seguit de càlculs matricials. Les matrius de translació i rotació tenen la forma següent:

$$\begin{pmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Exemple de càlcul matricial:

S'està escanejant una superfície plana. El punt on es vol portar l'extrem de l'escàner bé donat respecte un punt de la superfície. Per a trobar la matriu que s'ha d'enviar al robot, que defineix el desplaçament del terminal respecte la base del robot ( $\mathbf{M}_{rt}$ ), es coneixen les següents matrius, que s'il·lustren en la figura 1.6:

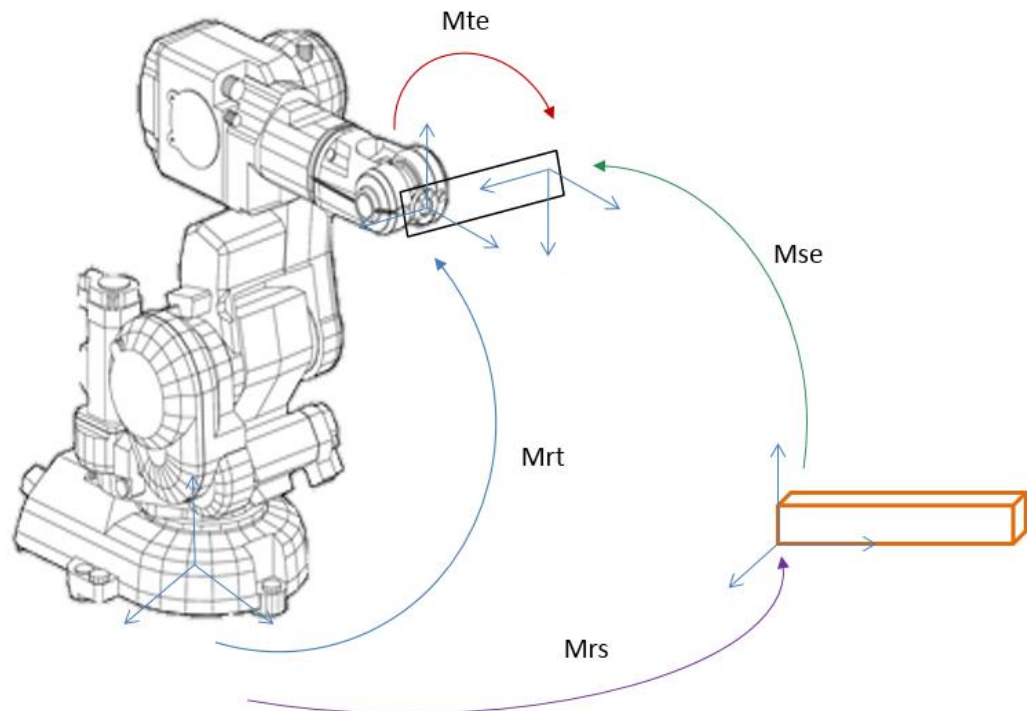
$\mathbf{M}_{se}$ : Desplaçament de l'extrem de l'escàner respecte la superfície.

$\mathbf{M}_{rs}$ : Desplaçament de la base de la superfície respecte a la base del robot.

$\mathbf{M}_{te}$ : Desplaçament de l'extrem de l'escàner respecte el terminal.

La matriu  $M_{rt}$  s'obté amb el següent càlcul:

$$M_{rt} = M_{rs} * M_{se} * (M_{te})^{-1}$$

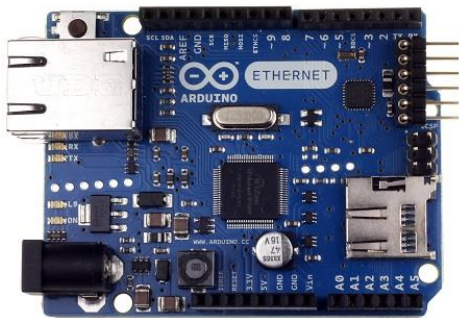


**Figura 1.6:** Explicació gràfica de l'obtenció de les matrius de desplaçament

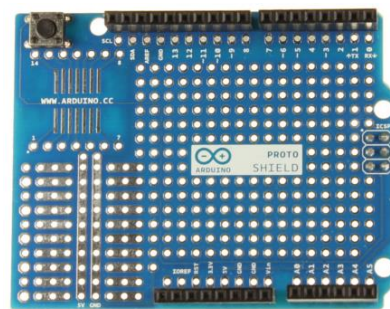
## 1.3. Placa microcontroladora

### 1.3.1. Hardware de la placa microcontroladora

La placa electrònica utilitzada és la *Arduino Ethernet Rev. 3* (figura 1.7). Està connectada a l'escàner, al làser i als sensors del marcadore. Tots els circuits i connexions s'han muntat en una placa *Proto Shield R3* (figura 1.8), que va acoblada a la primera.

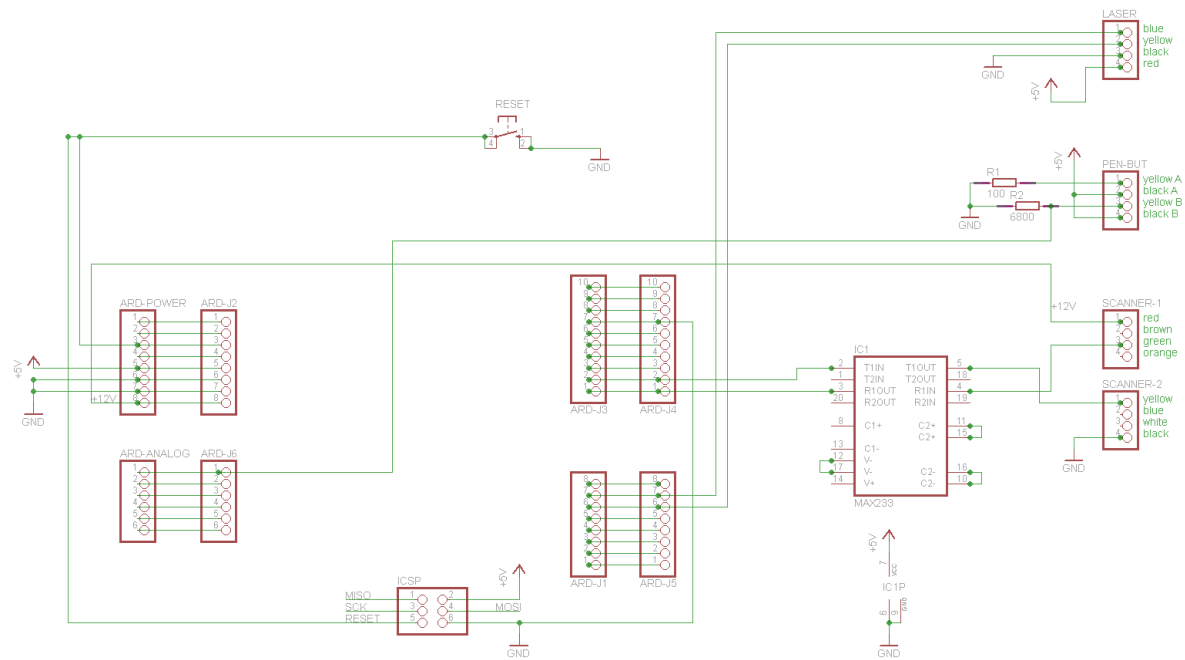


**Figura 1.7:** Arduino Ethernet Rev. 3

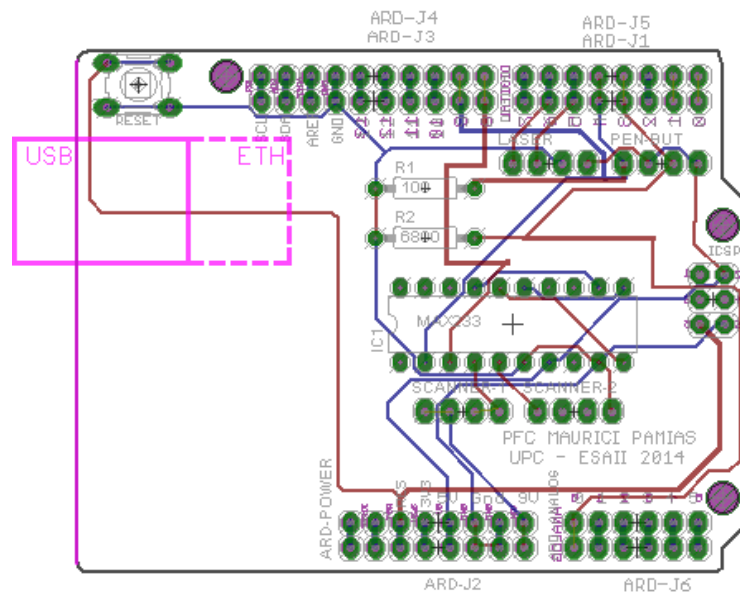


**Figura 1.8:** Proto Shield R3

Cada dispositiu té un circuit i uns components electrònics diferents que s'expliquen amb detall en els apartats corresponents. L'esquema del circuit final és el de les figures 1.9 i 1.10.



**Figura 1.9:** Schematic del circuit de la placa electrònica



**Figura 1.10:** Esquema del circuit de la placa electrònica

### 1.3.2. Comunicació entre l'aplicació i la placa microcontroladora

En aquest apartat s'explica l'estructura bàsica que segueix el programa que s'introdueix a la placa, i que té com a funció comunicar cada un dels dispositius amb el client que executa l'aplicació.

El codi està format per la declaració de variables i llibreries i dues funcions. La primera funció, anomenada *setup*, només s'executa un cop al iniciar el programa. En ella es crea un servidor amb una IP (que utilitzarà la aplicació principal per comunicar-s'hi), la connexió per port sèrie amb l'escàner, i s'inicialitzen els pins d'entrada i sortida. L'altra funció, anomenada *loop*, s'executa repetidament mentre la placa roman encesa. Aquesta funció s'encarrega de:

- Rebre i interpretar totes les informacions que arriben des de les connexions via Ethernet i sèrie.
- Comunicar-se amb els pins d'entrada i sortida.
- Confeccionar i enviar els missatges al servidor a través de Ethernet.

Els missatges que s'envien de l'aplicació al microcontrolador de la placa consten de dos elements. El primer és un caràcter que pot ser "0", "1", "2" i "3". Aquest codi indica la funció que es vol que faci l'Arduino.

0. Es demana informació sobre l'estat del marcador.
1. Es demana saber l'última lectura de l'escàner.
2. Activar el làser.
3. Desactivar el làser.

La funció *loop* conté subfuncions. Una d'elles s'encarrega de llegir els caràcters dels missatges que arriben des de la connexió via Ethernet. Aquesta funció llegeix un caràcter cada cop que s'executa. El primer caràcter que llegeix és aquest codi. Depenent de quin sigui el seu valor, executarà la funció corresponent. L'altre element del missatge és també un codi que s'ha de reproduir exactament igual en el missatge que es torna al client. Aquest codi és necessari ja que després l'aplicació principal la utilitza per saber quina funció ha d'executar a continuació.

El fragment de codi particular de cada funció s'explica amb detall en el capítol 2.

## Capítol 2

### Funcions del sistema desenvolupat

En aquest capítol s'explica en profunditat el funcionament de cada una de les tres funcions desenvolupades. També es presenta una quarta, que consisteix en apuntar a punts determinats de l'espai amb un làser. La utilitat d'aquesta funció no ha estat desenvolupada en aquest projecte, però sí que forma part del projecte EAPS.

De cada una de les funcions principals s'explica:

- Els components i el muntatge necessaris pel seu funcionament.
- L'estructura del software desenvolupat, que inclou:
  - Generació de la trajectòria del robot.
  - Control de moviment del robot.
  - Recollida de dades, només en el cas de l'escàner.
- Hardware i Software de la placa microcontroladora.

#### 2.1. Funció escaneig

La primera funció desenvolupada és l'escaneig. L'objectiu és poder escanejar objectes per obtenir-ne una imatge en 3D. Aquesta imatge, formada per un núvol de punts, ha de complir uns requisits de resolució per a que es pugui fer servir en la funció de polit.

Per obtenir aquesta imatge s'utilitza un escàner puntual. L'escàner, acoblat al capçal del robot, segueix una trajectòria determinada al voltant de la superfície. El robot s'atura cada certa distància, depenent de la resolució desitjada, perquè l'escàner pugui prendre les mesures.

S'han dissenyat tres mètodes diferents per escanejar un objecte. La utilització d'un o altre mètode depèn de la geometria de l'objecte que s'escaneja. En el primer, l'escàner segueix una trajectòria cilíndrica, en el segon una d'esfèrica, i el tercer segueix una trajectòria definida per un conjunt de triangles que defineixen la geometria de l'objecte.



### 2.1.1. Components de l'escaneig

Per a fer l'escaneig s'ha usat un escàner puntual de la marca *Acuity*, model AR-200-50. Aquest dispositiu projecta un feix de llum, creant un punt en la superfície. Aquesta llum és reflectida i captada per una càmera CMOS situada a dins de l'escàner. Aquesta càmera està a una certa distància de l'origen del làser, i, a partir de l'angle amb el que el feix de llum reflectit entra a la càmera, calcula la distància per triangulació. El rang de distàncies és de 16,6 mm fins a 67,4mm.

S'ha comprovat que aquest sensor pot donar dades errònies en canvis sobtats de profunditat i de material. També s'ha detectat un retard de l'ordre dels 200 milisegons des de que es llegeix la distància fins que la informació és enviada pel bus de dades.

També s'ha utilitzat components per muntar el circuit electrònic de la placa microcontroladora. Que s'explica amb profunditat a l'apartat 2.1.3.

### 2.1.2. Funcionament del procés d'escaneig

#### Generació de la trajectòria

Abans d'ordenar al robot que comenci l'escaneig s'ha de generar per anticipat, parcial o totalment, la trajectòria que seguirà. El càlcul de la trajectòria depèn del mètode d'escaneig escollit. A continuació s'expliquen els tres casos per separat:

#### Mètode 1: Trajectòria cilíndrica

En aquest mètode l'escàner segueix una trajectòria que cobreix part de la superfície de revolució d'un cilindre (Figura 2.1). És adequat per escanejar objectes que tinguin una geometria més o menys cilíndrica, com podria ser un ratolí d'ordinador, una llauna de refresc, etc.

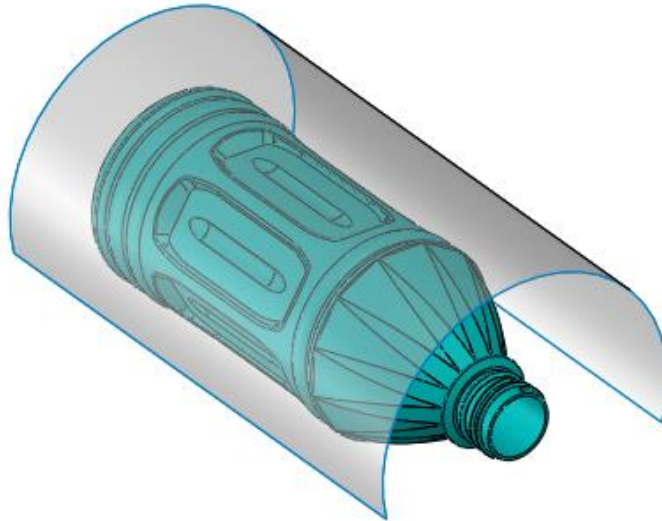
La trajectòria queda definida per un conjunt de punts a l'espai en els que l'escàner s'ha d'aturar a prendre les dades. Aquests punts formen unes línies paral·leles a l'eix del cilindre situades sobre la superfície. Per tant, per definir la resolució de l'escaneig, s'indica la separació en graus entre línies i la distància entre punts dins una mateixa línia

Per definir la trajectòria s'ha d'introduir un seguit de paràmetres a l'aplicació, que són:

- Alçada i radi del cilindre.
- Direcció i posició de l'eix del cilindre respecte el robot.
- Angles que defineixen la porció de superfície que es vol cobrir.
- Separació angular entre línies i separació entre punts d'una mateixa línia.

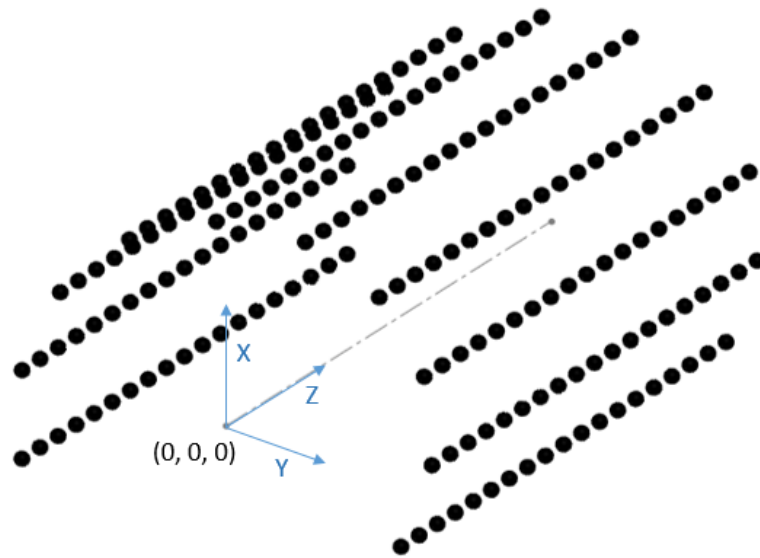
A l'hora de determinar el radi del cilindre s'ha de procurar que en tots els punts la distància entre l'escàner i la superfície de l'objecte estigui compresa dins el rang de l'escàner.

En aquest mètode els punts on s'ha de posicionar el robot es van calculant a mesura que el procés avança a partir dels paràmetres que s'han introduït a l'inici.



**Figura 2.1:** Superfície que cobreix la trajectòria de l'escàner

Les coordenades dels punts on s'ha d'aturar l'escàner estan definides per coordenades polars en els eixos X i Y i per cartesianes en el Z, respecte la base del cilindre. Les línies són paral·leles a l'eix Z, per tant tots els punt d'una mateixa línia tenen les mateixes coordenades X i Y, i estan separats entre ells en l'eix Z una distància igual a la resolució. Les coordenades queden definides com s'indica en les Figures 2.2 i 2.3.



**Figura 2.2:** Representació dels punts de trajectòria i l'origen de coordenades del cilindre

$$\begin{aligned} P_x &= \text{radi} * \sin(\text{alfa}) \\ P_y &= \text{radi} * \cos(\text{alfa}) \\ P_z &= \lambda \end{aligned}$$

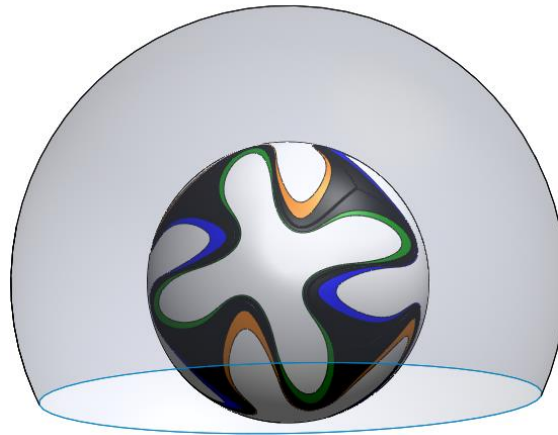
**Figura 2.3.** Coordenades dels punts segons el primer mètode

### Mètode 2: Trajectòria esfèrica

Aquest mètode s'utilitza per escanejar objectes que es puguin encabir dins un volum esfèric. En aquest cas la trajectòria que es segueix forma la superfície parcial d'una esfera, tal com es veu a la Figura 2.4.

Els paràmetres que s'han d'entrar a l'aplicació són:

- Diàmetre de l'esfera.
- Posició del centre respecte el robot.
- Angle a partir del qual es vol començar a escanejar.
- Separacions angulars dels dos angles que defineixen l'esfera.



**Figura 2.4:** Superfície que cobreix la trajectòria de l'escàner amb el mètode 2

$$\begin{aligned} P_x &= \text{radi} * (-\cos(\text{beta})) * \sin(\text{alfa}) \\ P_y &= \text{radi} * \cos(\text{alfa}) \\ P_z &= \text{radi} * \sin(\text{beta}) * \sin(\text{alfa}) \end{aligned}$$

**Figura 2.5:** Coordenades dels punts amb el mètode 2

La posició dels punts es dona en coordenades polars, i depèn de dos angles: l'alfa i el beta, tal com indica la Figura 2.5.

### Mètode 3: Definició d'una geometria genèrica basada en triangles

Amb aquest mètode, a diferència dels altres, es calculen i guarden en un fitxer de text els punts on s'ha de situar l'escàner per prendre les dades i abans de procedir a l'escaneig. La trajectòria que segueix el robot durant tot el procés queda definida abans de començar-lo.

La geometria de l'objecte a escanejar ve donada com un conjunt de triangles. S'usa aquest element geomètric ja que amb ell es poden representar totes les formes geomètriques possibles, amb més o menys resolució. Aquests triangles estan definits per tres punts (els tres vèrtex) donats en coordenades cartesianes. L'ordre amb què es donen aquests punts determina el sentit del vector normal a la superfície de l'objecte. A partir d'aquests triangles, s'executa una seqüència de funcions que dona com a resultat un fitxer que conté tots els punts de la trajectòria de l'escàner.

El primer pas és definir uns punts equidistants que segueixen el perímetre del triangle. La separació entre punts dependrà de la resolució d'escaneig desitjada. Des d'un dels tres vèrtexs cap als altres dos hi ha d'haver el mateix nombre de punts. Així, després d'aquest procés, els punts descriuen els triangles com indica la figura 2.6.



**Figura 2.6:** Generació de la trajectòria del mètode 3

En el següent pas es defineixen els punts interiors del triangle tenint en compte la trajectòria que seguirà el robot. Això es fa unint per parelles els punts dels dos costats del triangle que tenen el mateix nombre de punts amb línies de punts equidistants (Figura 2.7).



**Figura 2.7:** Generació de la trajectòria del mètode 3

Un cop fet això ja es té la trajectòria que seguirà en aquesta porció de superfície. Com que en la gran majoria dels casos, els objectes escanejats estan representats per més d'un triangle, s'ha de traçar també la trajectòria que porta el robot de l'últim punt de l'anterior triangle fins a l'inicial del següent.

Per garantir la seguretat de la trajectòria entre triangles es defineix un volum en forma de hexaedre que envolta la superfície. Cada cop que el robot s'hagi de desplaçar d'un triangle a un altre haurà de sortir del volum i tornar a entrar. Aquests punts també es desen en el fitxer que es llegirà posteriorment per moure el robot, però s'indiquen com a punts on no s'ha d'escanejar. Així, durant el procés, quan el robot arribi a un d'aquests punts es desplaçarà al següent sense consultar l'escàner ni desar cap dada.

## Execució de l'escaneig

### Mètode 1:

L'escàner es situa en el primer punt. Aquest és el que està a la coordenada  $Z=0$  de la línia que correspon l'angle inicial.

Per saber l'orientació de l'escàner se n'ha de determinar la direcció de l'eix  $z$ , que és el de sortida del làser. Es calcula a partir de les següents condicions:

1. Ha de ser perpendicular a l'eix del cilindre
2. Ha de passar pel punt i creuar-se amb l'eix.

Els altres dos eixos són irrellevants. Es calculen amb una funció que defineix primer l'eix  $x$  a partir de dues condicions: que sigui perpendicular a l'eix  $z$  i que el component paral·lel al pla horitzontal. Tot seguit el vector  $y$  es troba fent el producte vectorial del altres dos.

Un cop el robot ha arribat a la posició es pren la primera dada de distància respecte la superfície. Per a prendre les dades, es consulta la informació que arriba en aquell moment pel bus de dades corresponent a l'escàner. Com que l'escàner usat en aquest projecte té un retard no menyspreable, el robot ha d'estar parat una fracció de segon (de l'ordre dels 400 ms) abans no es llegeixi la dada.

Com s'ha comentat a l'apartat 2.1.1., l'escàner pot enviar dades errònies, que acostumen a indicar una distància més curta que la real. Per a minimitzar les conseqüències d'aquests errors es fan dues lectures per a cada posició amb un interval de 50 milisegons entre elles. D'aquestes dues lectures s'escull la que marca una major distància.

Un cop desada la distància en la variable corresponent, s'escriuen les dades desitjades en un fitxer. Per a cada punt es guarden:

- Les coordenades cartesianes i el quaternió l'element terminal en base robot.
- Les coordenades cartesianes i el quaternió de l'eina utilitzada.
- Els graus de gir de cada una de les 6 articulacions.
- La distància llegida de l'escàner.
- El punt resultant de calcular la mesura de l'escàner en referència base robot..

També es desa en un segon fitxer només les coordenades cartesianes del punt resultant. Aquest fitxer servirà per representar l'objecte escanejat en 3D.

Un cop desada la dada el robot es desplaça al següent punt de la fila. Quan s'acaba una fila el robot es desplaça fins el primer punt de la següent fila. Així fins que s'han recorregut totes les files.

### Mètode 2:

El segon mètode es desenvolupa d'una manera molt semblant al primer. La diferencia és la manera i l'ordre de donar les coordenades al robot.

En aquest cas, i com s'ha comentat, en l'apartat anterior, les coordenades són polars i depenen de dos angles: alfa i beta. El primer punt, per tant, correspon als dos angles inicials.

Per moure el robot al següent punt s'incrementa l'angle beta tant com s'ha indicat a resolució. S'incrementa beta fins que s'arriba al gir màxim d'aquest eix. A continuació es torna al gir inicial de beta, s'incrementa el gir d'alfa, i es torna a recórrer tot beta. La seqüència segueix fins que s'ha recorregut tot l'angle alfa.

### Mètode 3:

El procés pot començar un cop es tenen definits tots els punts per on haurà de passar l'escàner. El procediment és el mateix per cada un dels triangles que formen l'objecte. A continuació s'explica la seqüència d'escaneig per dos triangles.

Després de donar l'ordre d'escanejar, el robot es situa en el punt inicial del primer triangle. Per aproximar-se a aquesta posició inicial, es col·loca primer a una distància determinada en direcció contrària a la superfície, per assegurar d'aquesta manera que el robot no xoqui amb l'objecte en l'aproximació. Un cop allà, el robot s'apropa a la superfície fins que arriba a la posició on l'escàner ha de prendre la primera dada.

Les dades es prenen i es desen en els fitxer de la mateixa manera en el mètode 1.

Després de desar les dades, el robot es desplaça al segon punt i segueix el mateix procediment per a llegir la distància i posteriorment guardar les dades, i així successivament.

Un cop escanejats tots els punts d'un triangle el robot s'allunya de l'objecte fins a una posició segura i es desplaça fins el punt inicial del següent triangle.



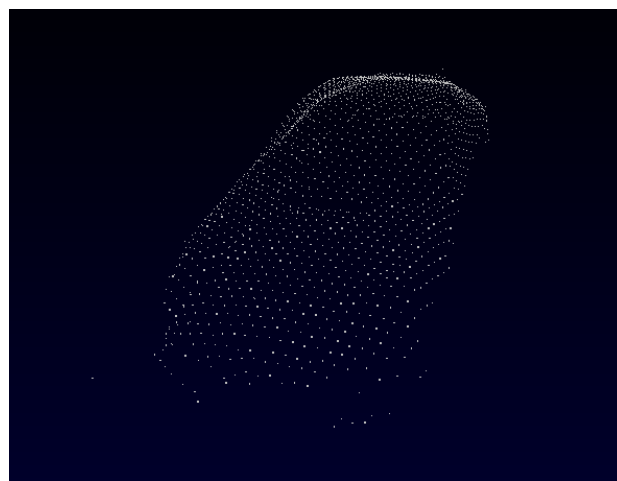
## Tractament de les dades obtingudes en l'escaneig

Els punts obtinguts en l'escaneig han de ser tractats per a poder ser aptes pel procés de polit. Aquest tractament es fa amb Excel. S'importen al programa les dades obtingudes a l'escaneig. Mitjançant la representació amb gràfics de la coordenada Z en funció de les coordenades X i Y es detecten els punts que es diferencien en excés dels que els envolten i s'eliminen de la llista.

En l'escaneig d'un model en miniatura de cotxe (Figura 2.8) s'ha obtingut el núvol de punts de la Figura 2.9.



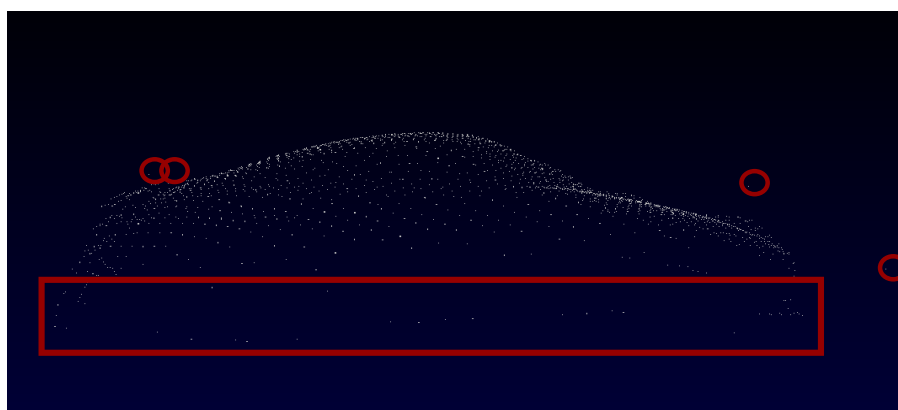
**Figura 2.8:** Model en miniatura de cotxe



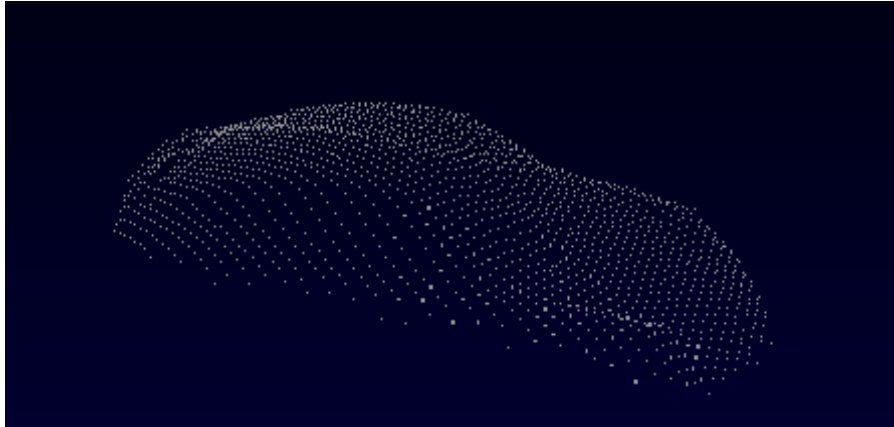
**Figura 2.9:** Núvol de punts obtinguts en l'escaneig

En la figura 2.10 es mostren els punts que s'eliminen en el tractament de dades. La banda de punts inferior s'elimina per les limitacions de profunditat que presenta el procés de polit. Els altres punts són dades errònies que no s'han detectat i eliminat en l'escaneig.

Després de tractar les dades s'obté el núvol de punts de la Figura 2.11, el qual ja és apta per ser introduït en el procés de polit.



**Figura 2.10:** Punts eliminats en el tractament de dades



**Figura 2.11:** Núvol de punts utilitzat en el polit

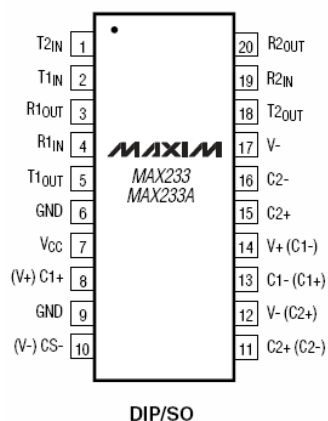
### 2.1.3. Electrònica del procés d'escaneig

Originalment l'escàner puntual es connecta a l'ordinador via port sèrie. La connexió és fa a través d'un cablejat molt feixuc que en ocasions pot dificultar la mobilitat del robot. Per això s'ha optat per tallar el cable i connectar-lo a la placa microcontroladora, que es troba al mateix capçal del robot. D'aquesta manera s'aconsegueix un espai de treball més net i una comunicació més clara amb el dispositiu.

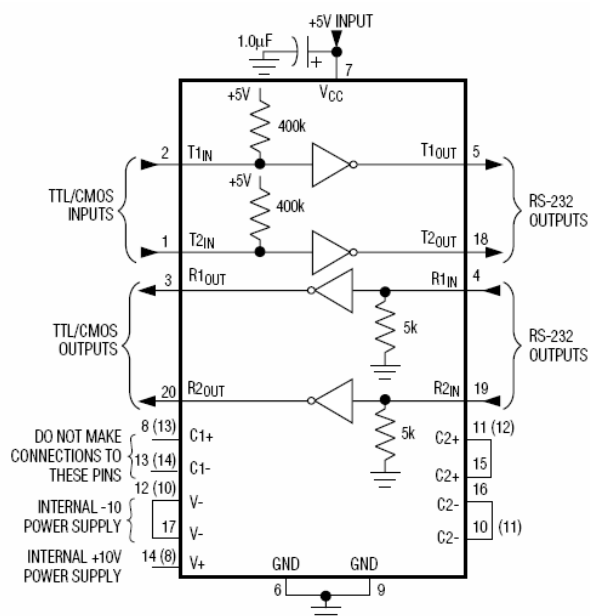
#### Hardware de la placa microcontroladora de la part de l'escàner

L'escàner envia la informació a través d'un port sèrie. Dels 8 cables que ens surten, només interessen els de les dades Rx i Tx, el d'alimentació i el de contacte a terra. Com que la placa Arduino és alimentada amb una font de +12V, el cable d'alimentació de l'escàner, de color vermell, es connecta al piu corresponent a la tensió d'alimentació de la placa. De la mateixa manera el cable de contacte a terra de l'escàner, de color negre, es connecta al GND de l'Arduino. En canvi els cables que transfereixen les dades Rx i Tx no es poden connectar directament als seus homònims de la placa Arduino, ja que l'escàner treballa a una tensió de  $\pm 12V$ , mentre que la placa ho fa a +5V. Per a solucionar-ho s'ha utilitzat el circuit integrat *MAX233cpp*. Connectant els seus pins de la manera que s'explica a continuació es transforma la tensió de transmissió de bits de  $\pm 12V$  a +5V.

Les estructures interna i externa del xip s'il·lustren en les Figures 2.12 i 2.13.

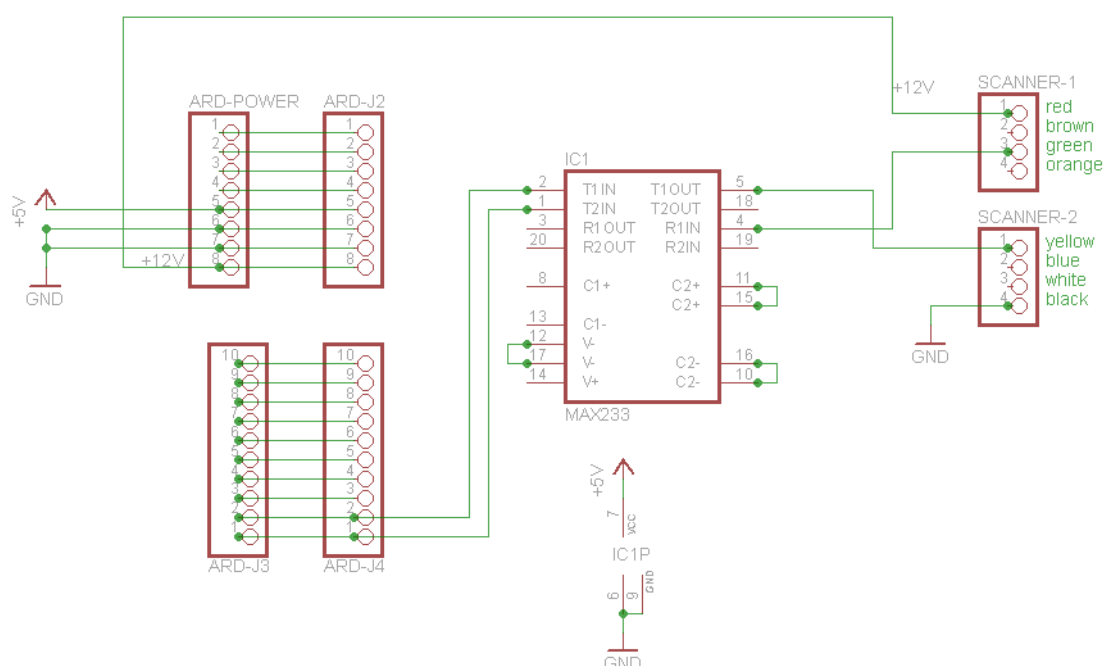


**Figura 2.12:** Estructura externa del xip MAX233



**Figura 2.13:** Circuit integrat del xip

El microxip permet la transformació de dos canals de dades. En aquest cas només en cal un, per això es deixen lliures els pins 1, 18, 19 i 20. Les connexions entre escàner, microxip i la placa queden definides en l'esquema de la Figura 2.14.



**Figura 2.14.** Schematic del circuit integrat a la placa Arduino

### Software de la placa microcontroladora de la part d'escàner

Com s'ha explicat en l'apartat 1.5, el microcontrolador reconeix que se li demana la lectura de l'escàner quan el codi que rep és "1". Un cop llegit el codi, en les següents iteracions de la funció *loop* l'Arduino prepara i envia el missatge de retorn, que conté la dada de distància i un codi que, com ja s'ha explicat, ve donat pel missatge que es rep del servidor.

El programa intern de la placa actualitza constantment la variable distància. Ho fa en una subfunció dins el *loop*. Aquesta subfunció llegeix els caràcters que arriben des de la connexió amb l'escàner via port sèrie. La connexió es fa a través dels pins 8 i 9 de la placa, que fan de receptor i transmissor respectivament, gràcies a una funció anomenada *SoftwareSerial*.

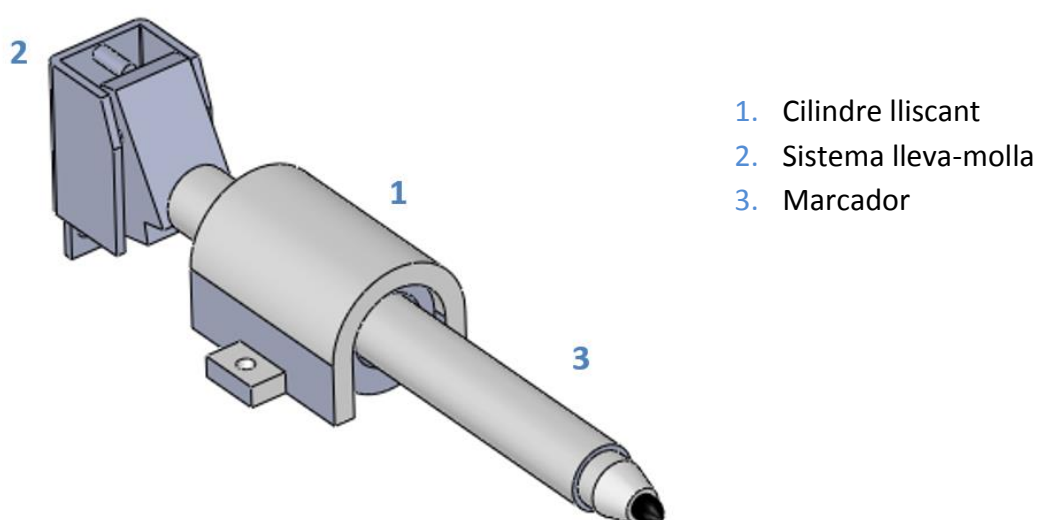
Per cada dada de distància l'escàner envia entre 5 ó 6 caràcters: dues o una xifra (depèn de si la distància és més gran o més petita que 10mm), un punt i tres xifres decimals. A mesura que arriben les dades s'emmagatzemen en una variable tipus *String*. Quan el caràcter que arriba és un ".", s'inicia un comptador (amb valor inicial 0), que de valor en una unitat per cada caràcter de més que es llegeix. Quan el comptador arriba a 3, vol dir que s'ha llegit l'últim dels decimals. En aquest moment es desa el que hi ha a la variable en una nova variable i, perquè es pugui començar a emmagatzemar els caràcters de la següent dada, se'n borra el contingut. Aquesta nova variable, és la que s'envia en el missatge de retorn juntament amb el codi anteriorment comentat.

## 2.2. Funció dibuixar

En aquesta funció el robot ha de ser capaç de representar un dibuix en una superfície. Per a fer-ho s'ha instal·lat un marcador en el capçal amb un muntatge que, com s'explicarà més extensament a continuació, permet saber si s'està dibuixant o no. El dibuix ve determinat per un conjunt de punts a l'espai. Unint els punts en l'ordre en que són donats queda representat el traç. Per a cada punt també s'indica el vector normal al pla. Aquesta dada és necessària, ja que conèixer-la permet aproximar el marcador a la superfície amb la inclinació desitjada, que en la majoria dels casos serà la mateixa normal. És possible que un dibuix estigui format per traços separats.

### 2.2.1. Mecanisme del marcador

En el procés de dibuixar, la punta del marcador ha d'estar en contacte amb la superfície, a més d'aplicar-hi una certa pressió. Per assegurar que aquesta aproximació es faci correctament, s'ha optat per afegir un mecanisme que permeti un moviment lineal del marcador en la direcció del seu eix. El mecanisme (Figura 2.15) està format per un cilindre lliscant per la seva cara interna, que és el que subjecta el marcador, i un sistema lleva-molla que aplica una força a la part posterior del marcador. D'aquesta manera el que es busca és que el marcador estigui una mica desplaçat cap a dins mentre es trobi dibuixant.



**Figura 2.15:** Components i mecanisme del sistema demarcat

Evidentment, seria contraproduent que la aproximació fos massa gran, ja que la lleva té un recorregut de, aproximadament 1cm. És per això que s'ha afegit un sistema de sensors per, com s'ha mencionat en l'apartat anterior, controlar la posició en que es troba la lleva. El recorregut del marcador al llarg de la guia correspon amb l'angle de gir que té permès la lleva. Per dibuixar en bones condicions, la lleva s'ha de trobar en un punt entremig entre els angles màxim i mínim. Per aquesta funció s'han estudiat i provat dos sistemes diferents, explicats a continuació.

#### Sistema 1: Sensors de contacte

Aquest mètode consisteix en situar dos sensors de contacte a la peça metàl·lica que subjecta la lleva i la molla. Els dos sensors són petits interruptors accionats per un botó. Quan es pitja el botó, l'interruptor es tanca. Un dels sensors fa de topall de la lleva quan no hi ha contacte, situació en la qual l'interruptor està tancat. L'altre, en canvi, fa de topall en el cas oposat, és a dir, quan el marcador ha arribat al final del recorregut. Un senzill circuit electrònic connectat a la placa Arduino permet tenir un cert control sobre la posició de la lleva. Si el primer interruptor està tancat vol dir que el marcador no té contacte amb la superfície. Si els dos interruptors estan oberts la lleva està en un punt entremig del recorregut, fet que indica que hi ha el contacte desitjat per dibuixar. Si és el segon interruptor el que està tancat, el contacte és massa gran i el marcador s'ha de retirar.

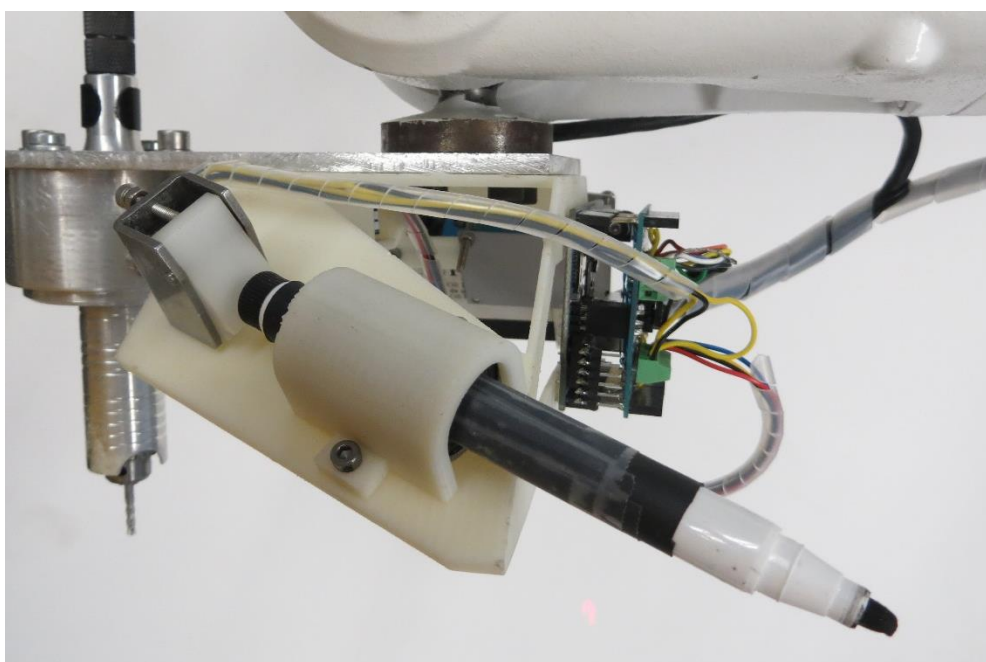
#### Sistema 2: Sensor lumínic

Aquest altre mètode és un sensor lumínic que permet calcular la distància amb un objecte en un rang de 1 a 8 mm. El sensor es pot dividir en dos dispositius. Un és un led que envia un feix de llum, i l'altre un transistor que deixa passar el corrent quan capta part d'aquesta llum. Si no hi ha cap objecte en el rang de distàncies del sensor, el receptor no capta la llum, i el transistor està tancat. En el moment en que un objecte es posiciona a 8 mm de distància, la corrent comença a passar, i augmenta a mesura que l'objecte s'aproxima al mil·límetre de distància. Gràcies a un senzill circuit electrònic es pot transformar aquest corrent elèctric en un voltatge, que oscil·la entre 0 i 5 V. Aquesta tensió és la dada que llegeix la placa i la que es fa servir per calcular la distància que llegeix el sensor.

El sensor es col·loca a la peça metàl·lica, davant de la part inferior de la lleva. Així, quan el marcador no està en contacte amb la superfície, la distància que es llegeix és

màxima, i es va reduint a mesura que el marcador es desplaça cap enrere. D'aquesta manera hi ha un rang de valors de la distància llegida en els quals el marcador està en la posició adequada per dibuixar.

Finalment l'opció escollida ha estat la del sensor lumínic, ja que presentava dues avantatges molt rellevants respecte l'altra. La primera és que el muntatge que s'havia de fer amb el sistema dels botons requeria que el marcador hagués de fer molta pressió contra la superfície per desplaçar-se. En canvi el sistema escollit utilitza una molla més de constant elàstica més baixa, que presenta menys resistència al desplaçament. L'altra avantatge és que amb el sensor lumínic aporta més informació que els botons, ja que permet saber en tot moment la posició exacte de la lleva. Aquesta informació es pot fer servir per millorar el procés de marcat.



**Figura 2.16:** Fotografia del mecanisme de marcat muntat al capçal

## 2.2.2. Funcionament del procés marcat

### Generació de la trajectòria

Com s'ha comentat en la introducció d'aquest apartat el dibuix ve determinat per un conjunt de punts. Aquests punts es donen amb un fitxer que, per cada punt, aporta les dades com en la Taula 2.1.

Índex	X (mm)	Y (mm)	Z (mm)	Vx	Vy	Vz
0	150.0	-40.3	16.5	0	0	-1

**Taula 2.1:** Estructura de les dades que utilitza la funció de marcat

Primer es dona l'índex que, en el cas de que hi hagi més d'un traç, indica a quin pertany cada un dels punts. Després es donen les coordenades cartesianes del punt i per últim el vector normal de la superfície. Abans de començar el procés, es llegeix el fitxer i es guarden les coordenades i el vector de cada punt en una matriu. També es crea una segona matriu que indica el primer i últim punt de cada traç.

Com s'ha comentat en l'apartat 1.2., el recorregut que fa el robot entre dos punts és imprevisible. Això és un handicap a l'hora de dibuixar, ja que el recorregut és de màxima importància. Per a solucionar això s'utilitza la solució també comentada en el mateix apartat. Per a tal propòsit s'ha creat una funció anomenada *PuntsIntermitjos*, que s'executa cada cop que es comença a dibuixar un traç. Els *inputs* de la funció són: la separació màxima que es desitja que tinguin els punts consecutius, la matriu que té indexats tots els punts, i la matriu on s'han desat les coordenades dels punts. En aquesta funció es crea una nova matriu on s'hi troben tots els punts d'aquest traç i els punts entremetjats necessaris, tots ells en l'ordre que ha de seguir el robot a l'hora de dibuixar.



## Execució del procés de marcat

El procés de dibuixar s'inicia portant el robot a la posició de seguretat del primer traç. La posició de seguretat està situada a una certa distància, en direcció a la normal de la superfície, del primer punt a dibuixar.

Tot seguit s'acosta el marcador cap a la superfície. Cada certa distància, de l'ordre del mil·límetre, es comprova, gràcies al sistema explicat en l'apartat 3.1, si el marcador està fent contacte amb la superfície. En el moment en que aquest contacte existeix, el robot para d'aproximar-se a la superfície i es mou, en direcció paral·lela a la superfície, al següent punt que indica la matriu. Com es presumible, durant aquest moviment el marcador haurà dibuixat una recta entre ambdós punts.

Un cop s'ha arribat al nou punt es torna a comprovar el desplaçament del marcador. Aquest ha d'estar dins un rang de valors. No convé ni que el desplaçament sigui molt petit, ja que es podria perdre el contacte amb la superfície, ni que sigui massa gran, ja que hi podria haver col·lisió entre el marcador i el sensor. Si es dóna el primer cas el robot s'aproxima a la superfície, i si es dóna el segon se n'allunya, fins que el desplaçament és adequat. A continuació el robot es mou al següent punt de la llista.

Aquest procés segueix fins que s'acaben els punts d'un traç. En el cas de que se n'hagi de dibuixar un altre, el robot es situa a la posició de seguretat d'aquest, i torna a realitzar el mateix procés.

En el diagrama de la Figura 2.17 s'explica gràficament el procediment.

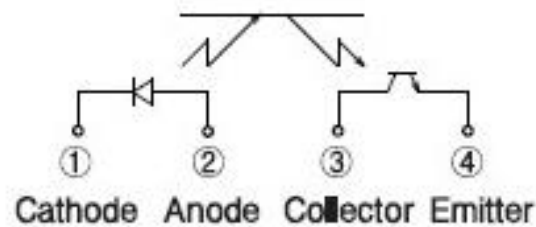


### 2.2.3. Electrònica del marcat

#### Hardware de la placa microcontroladora de la part del marcat

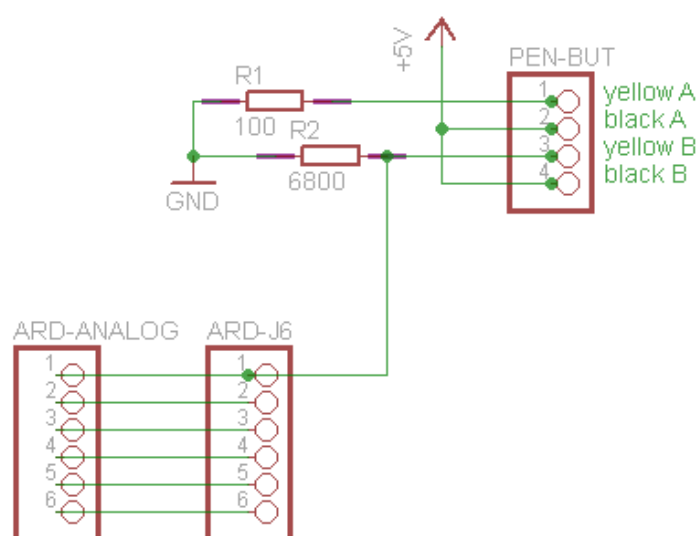
Els components necessaris pel sistema escollit són:

- El sensor lumínic, model sg-2bc (Figura 2.18).
- Una resistència de 100 ohms.
- Una resistència de 6800 ohms.



**Figura 2.18:** Esquema del funcionament del sensor sg-2bc

L'esquema d'aquesta part del circuit és el de la Figura 2.19.



**Figura 2.19:** Schematic de la part del circuit del marcadore

Com es veu en l'esquema el sensor té quatre pins. Els pins 1 i 3 es connecten a 5V. El pin 2 es connecta un terminal de la resistència de 100 ohms. El 4 es connecta a l'altra resistència. Els terminals lliures de les dues resistències es connecten a terra. La tensió del pin 4 és la que indica la distància que llegeix el sensor, així que es connecta amb el pin l'A0 de la placa, que és analògic.

### **Software de la placa microcontroladora de la part del marcat**

Com s'ha comentat en l'apartat 1.5. el codi que correspon a aquesta funció és el "0". Un cop l'Arduino rep aquest caràcter del servidor, llegeix la dada del pin analògic A0 i en desa el valor en una variable.

De la mateixa manera que en la funció de l'escàner, durant les següents iteracions de la funció *loop* el programa llegeix la resta de missatge provinent del servidor, on hi ha el codi que s'ha de retornar. El missatge que s'envia de tornada conté la dada llegida, un espai, i el codi esmentat.

## 2.3. Funció de polit

El procés de polit utilitza les dades obtingudes en l'escaneig de l'objecte per reproduir-lo en un motlle eliminant-ne el material sobrant. L'objectiu és obtenir una rèplica el més exacte possible tenint en compte certes limitacions.

- El diàmetre de la broca limita la resolució del resultat.
- La broca té una longitud de 20 mm. S'ha de tenir això en compte a l'hora de polir a més profunditat perquè no hi hagi col·lisions entre l'eina i la superfície.
- S'ha auto-imposat, per simplificar el codi de programació, que l'eix de la broca sigui sempre vertical.

Algunes d'aquestes limitacions són fruit de les característiques dels materials amb el que es treballa. D'altres són degut a un temps limitat per al desenvolupament del software. Amb una dedicació exclusiva es pot desenvolupar un software més avançat, amb el que es podria reproduir objectes amb més precisió i velocitat, però queda fora l'abast d'aquest treball.

### 2.3.1. Components

Els components necessaris en aquesta funció són:

- Sistema multi-eina Dremel.
- Motlle.

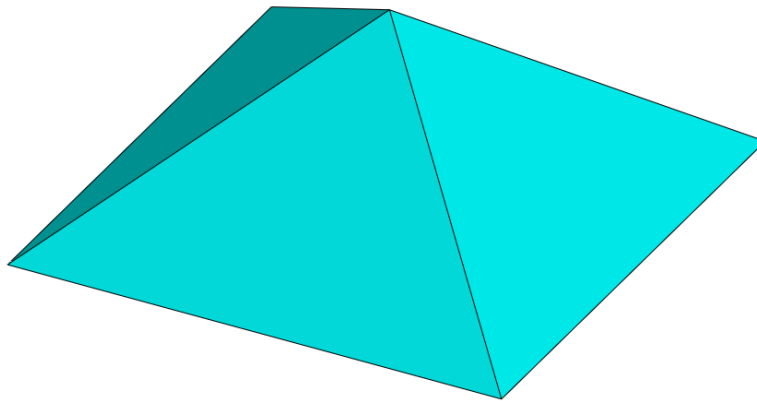
El sistema multi-eina inclou el motor, l'eix flexible i les broques. El motor elèctric dóna parell i velocitat de gir, que es transmeten a través de l'eix flexible fins el capçal, que és on s'instal·la l'eina que es vol utilitzar. Per aquesta funció s'utilitzen broques capaces de polir tant amb la base com amb el lateral. Per aquest treball s'ha utilitzat una de 60 mm de longitud i 3 mm de diàmetre.

El motlle ha de complir dues condicions:

1. Ha de ser més gran que la rèplica a fabricar.
2. Ha de ser d'un material prou tou com perquè la broca no es deteriori i el robot no hagi de fer sobreesforços.

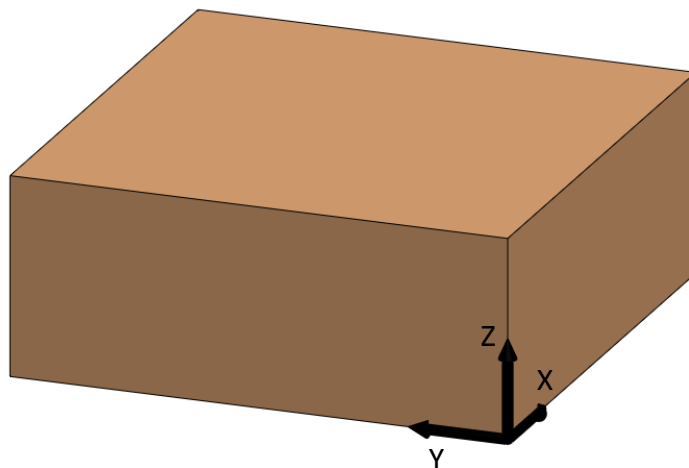
### 3.2.2. Funcionament del procés de polit

En aquest apartat s'explica gràficament el procés que s'ha de seguir per reproduir un objecte com el de la Figura 2.20.



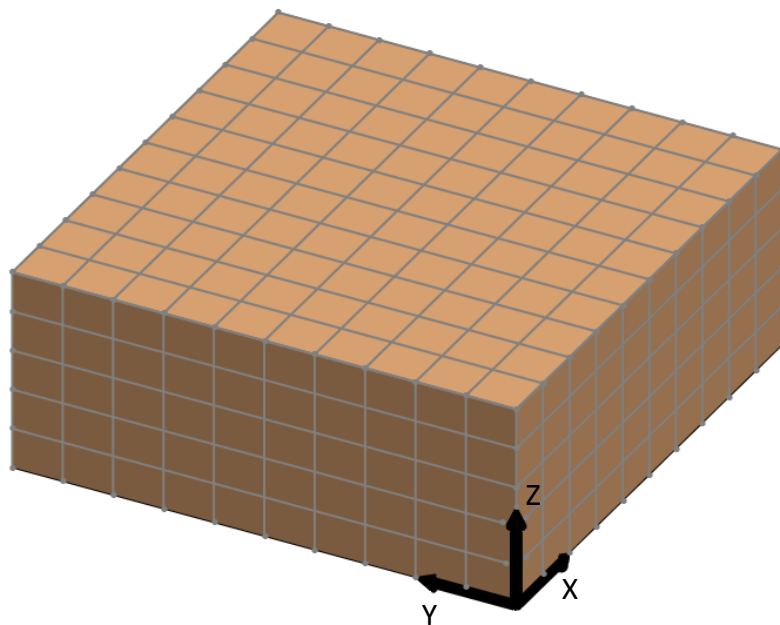
**Figura 2.20:** Objecte a reproduir per la funció de polit

1. El primer pas és escanejar l'objecte per obtenir les dades necessàries del seu contorn. Aquestes dades, si cal, han de ser tractades perquè siguin aptes pel procés de polit.
2. S'escull un motlle que cobreixi tot el volum de l'objecte escanejat (Figura 2.21).



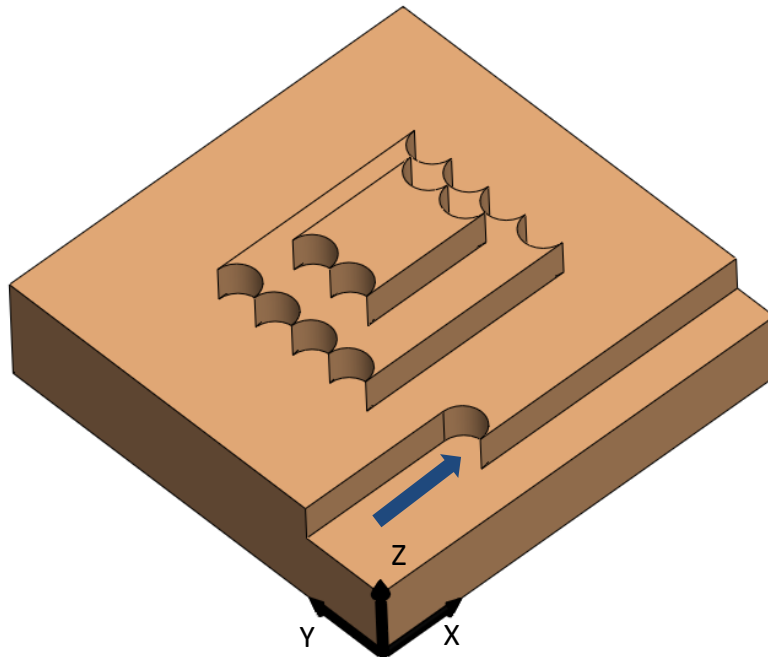
**Figura 2.21:** Motlle abans de començar el procés

3. Es divideix el pla que formen els eixos X i Y del motlle en quadrícules. Aquestes són més grans o petites depenent de la resolució desitjada.
4. Es divideix també l'eix vertical en capes de manera que queda tot el volum dividit en hexàedres regulars, com es veu a la Figura 2.22.



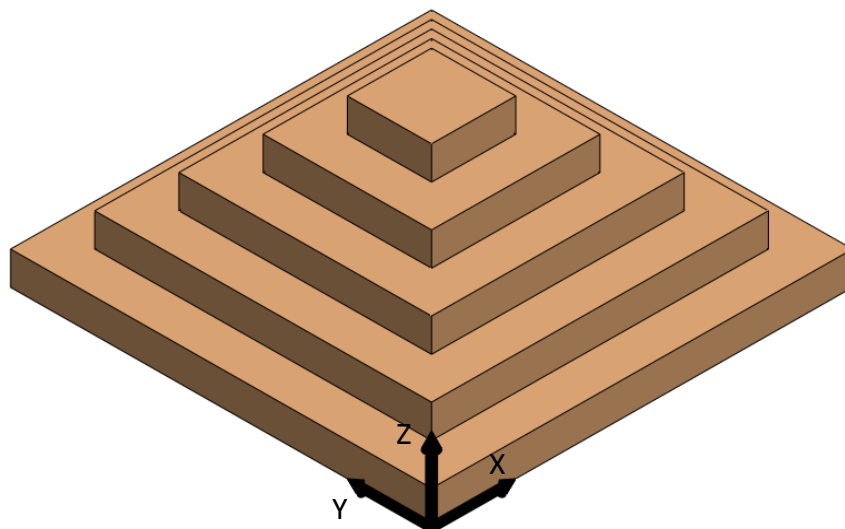
**Figura 2.22:** Divisió del volum del motlle en sub-volums hexaèdrics

5. Es poleix capa per capa, començant per la superior. A cada capa s'eliminen les quadrícules que, segons el fitxer de dades, no contenen material. El polit es fa, fila per fila, paral·lel a l'eix X, tal com s'observa a la Figura 2.23.



**Figura 2.23:** Estat del motlle durant la primera passada del procés

6. Un cop acabades totes les capes es torna a repassar tota la peça des del principi, però, en comptes de fer-ho paral·lelament a l'eix X, es fa paral·lel a l'eix Y. D'aquesta manera s'eliminen les imperfeccions en els contorns provocades per la forma cilíndrica de la broca que es poden observar a la Figura 22. El resultat final és el de la Figura 2.24.



**Figura 2.24:** Rèplica de l'objecte obtinguda amb el polit



## Generació de la trajectòria

Abans d'executar el programa, i com es fa en les altres funcions, s'ha de definir la trajectòria que seguirà el robot. El fitxer de dades que arriba de l'escaneig és un núvol de punts 3D. El primer que es fa és desar el llistat de punts en una matriu de 3 columnes i les files necessàries.

Es crea un volum virtual igual que el motlle, amb l'origen de coordenades en un dels vèrtexs. Es divideixen els tres eixos X, Y i Z en files, columnes i capes respectivament. El nombre de divisions ve determinat per la dimensió del motlle en aquell eix i la resolució desitjada en el model final. Per exemple, si l'alçada del motlle és de 500 mil·límetres i es vol una resolució en l'eix z de 2,5 mil·límetres, s'ha de dividir el volum en dues-centes capes. D'aquesta manera el volum queda dividit en quadrícules. Cada una està definida per tres índexs, que indiquen quina és la seva fila, columna i capa.

Tot seguit, solapant el volum virtual i el núvol de dades, es mira, capa per capa, quines quadrícules del volum contenen un o més punts del núvol. En una nova matriu es desen, per cada capa, els índexs de les quadrícules que contenen punts i que, per tant, no s'hauran d'eliminar.

## Execució del polit

Un cop tractades les dades ja es pot començar amb el procés de polit. Aquest es fa per capes, i és igual per cada una d'elles. A continuació s'explica el procés per una capa.

Per cada capa el robot ressegueix totes les quadrícules per files. Comença recorrent totes les columnes de la fila 1, després totes les de la fila 2, i així fins que ha cobert tota la superfície. Com que el diàmetre de la broca és, en la majoria dels casos, inferior a l'amplada de les files, s'ha de fer tantes passades a una mateixa fila com calguin per cobrir-la tota.

Es comença per situar l'extrem de la broca a sobre la cantonada inicial de la quadrícula corresponent a la primera fila i columna. Tot seguit es consulta en la matriu que conté els índexs, explicada a l'apartat anterior, si el material d'aquella quadrícula ha de ser eliminat o no. En cas afirmatiu, el robot descendeix fins a l'altura inferior de la capa. En l'altre cas es queda a la mateixa altura i es desplaça fins situar-se a sobre de la primera quadrícula que s'ha de polir.

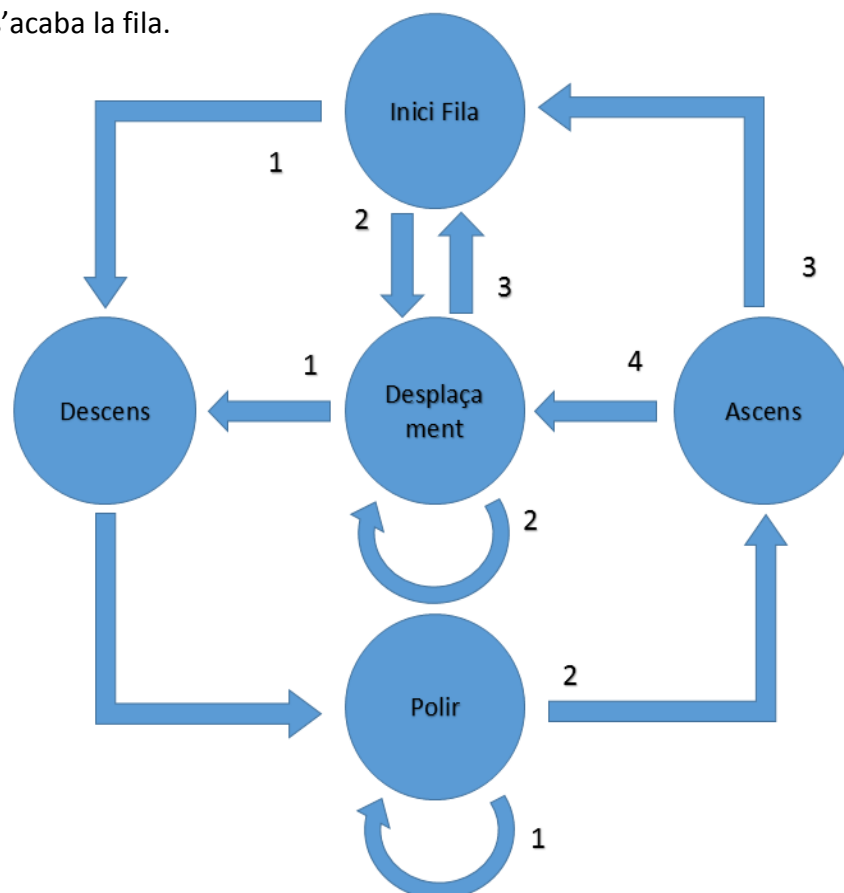
A continuació el robot es desplaça en la direcció de la fila fins que ha recorregut tota la quadrícula. Seguidament comprova si la següent quadrícula ha de ser o no eliminada. En cas afirmatiu, segueix polint en la mateixa direcció.

Segueix així fins que topa amb una quadrícula que no ha de ser eliminada. En aquest cas la broca puja fins situar-se a una altura superior al motlle. Tot seguit comprova quina és la següent quadrícula que ha de ser polida i s'hi situa al damunt, per posteriorment descendir i tornar a polir. El funcionament s'explica amb el diagrama de la Figura 2.25.

Un cop s'ha acabat de polit una capa es fa el mateix amb la següent, i així fins que s'han acabat totes.

Transicions:

- 1) La següent quadrícula s'ha d'eliminar.
- 2) La següent quadrícula no s'ha d'eliminar.
- 3) S'acaba la fila.
- 4) No s'acaba la fila.

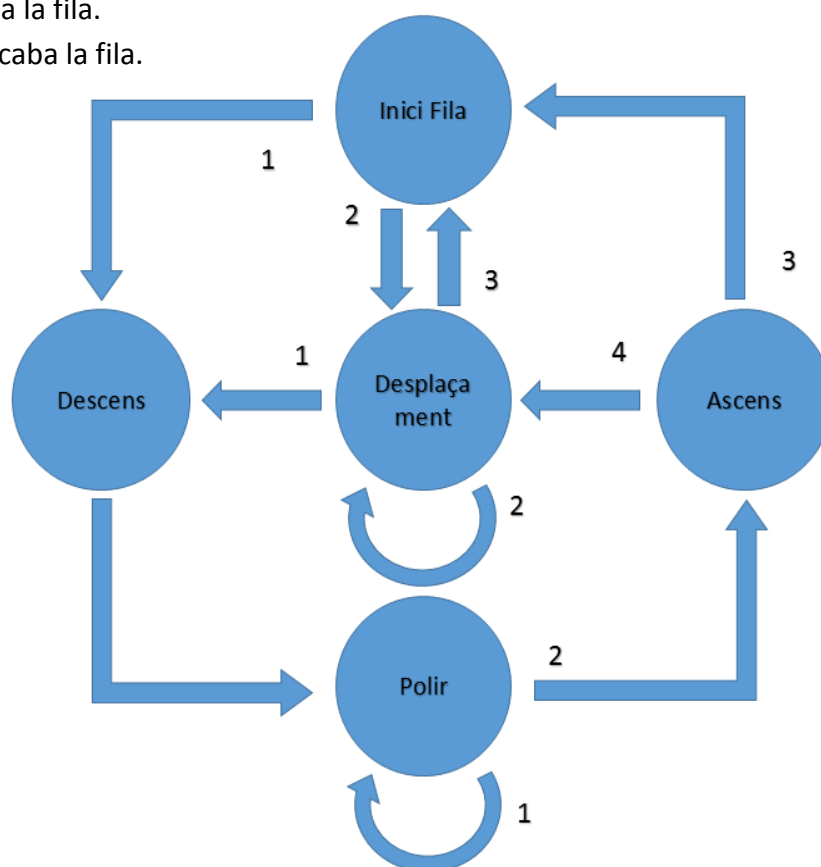


**Figura 2.25:** Diagrama d'estats de la primera passada

Com s'ha comentat anteriorment en aquest apartat, el resultat final té un mal acabat en les cantonades per culpa de la forma cilíndrica de la broca. Per solucionar això parcialment es fa una segona passada per eliminar el material sobrant. En aquesta segona passada la broca recorre les quadrícules en la direcció de les columnes, al contrari que abans, poleix les quadrícules segons els criteris que es veuen en el diagrama de la Figura 2.26.

Transicions:

- 1) La quadricula ha estat polida i es compleix una de les dues següents:
  - a. Alguna de les quadricles que té al costat de la mateixa fila no s'han eliminat.
  - b. La següent quadrícula també ha estat polida i alguna de les dues que té al costat no.
  - c. L'anterior quadrícula ha estat polida en aquesta segona passada i alguna de les dues que té al costat conté material.
- 2) No es compleixen les condicions de la transició 1.
- 3) S'acaba la fila.
- 4) No s'acaba la fila.



**Figura 2.26:** Diagrama d'estats de la segona passada

La solució és parcial perquè les cantonades exteriors no es poden repassar, ja que s'hauria d'eliminar material de la peça final.

Aquesta segona passada té una conseqüència negativa, i és que la resolució en la direcció de les columnes té un valor mínim igual al diàmetre de la broca, cosa que fins ara només passava amb la resolució en la direcció de les files.

## 2.4. Funció apuntar

Com s'ha comentat a la introducció, l'objectiu d'aquest projecte és simular a petita escala el que es vol aconseguir amb el projecte EAPS.

El projecte EAPS desenvolupa un sistema d'escanejat i polit de grans superfícies. Per cobrir superfícies de grans dimensions el robot es mou al llarg d'una guia. És molt important saber amb gran exactitud la posició del robot respecte la superfície. Com que el robot es mou al llarg de la via, és possible perdre'n la referència. Per això s'ha desenvolupat un sistema per poder situar el robot respecte una base fixe en tot moment.

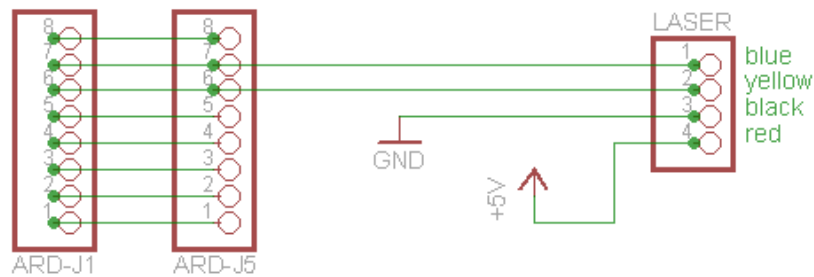
Aquest sistema consisteix en dues càmeres col·locades una al costat de l'altre en un lloc de l'entorn de treball on el robot hi pugui apuntar amb un làser des de qualsevol posició. Les càmeres tenen al davant una superfície plana semitransparent, de tal manera que si el làser apunta a la zona que entra dins el camp de visió de la càmera, aquesta detecta el punt i li assigna unes coordenades.

Un cop s'ha trobat les dues orientacions que serveixen per apuntar a les càmeres, s'utilitzen per situar la base del robot respecte la base fixe amb una senzilla triangulació.

Com que per aquest projecte el robot es manté fixe respecte al terra en tot moment no s'ha desenvolupat aquesta funció. Tot i així s'ha tingut en compte, i s'ha incorporat el làser en el capçal, s'ha fet un circuit electrònic i s'ha programat l'Arduino per tal que es pugui apagar i encendre el làser des d'una aplicació.

## Programació i electrònica de la placa electrònica pel làser

Del làser en surten quatre cables: el d'alimentació (5V), el de contacte a terra, i dos cables més (blau i groc). La funció del cable groc és la modulació en tren de pulsos (PWM). En aquest projecte no és necessària la modulació, i es manté per aquest cable una sortida fixa a 5V. El cable blau és el que habilita o des-habilita el làser, depenent de si rep o no una tensió de 5 V. Per això s'ha connectat a un piu de sortida digital, que es posa a 5 V quan es vol el làser encès, i a 0V quan ha d'estar apagat. L'esquema del circuit és el de la Figura 2.27.



**Figura 2.27:** Schematic del circuit de la part de làser de l'Arduino

Per comunicar a l'Arduino que es vol encendre el làser, se l'hi ha d'enviar un missatge amb el codi "2". Per indicar que es vol apagar el codi és el número "3".



## Capítol 3

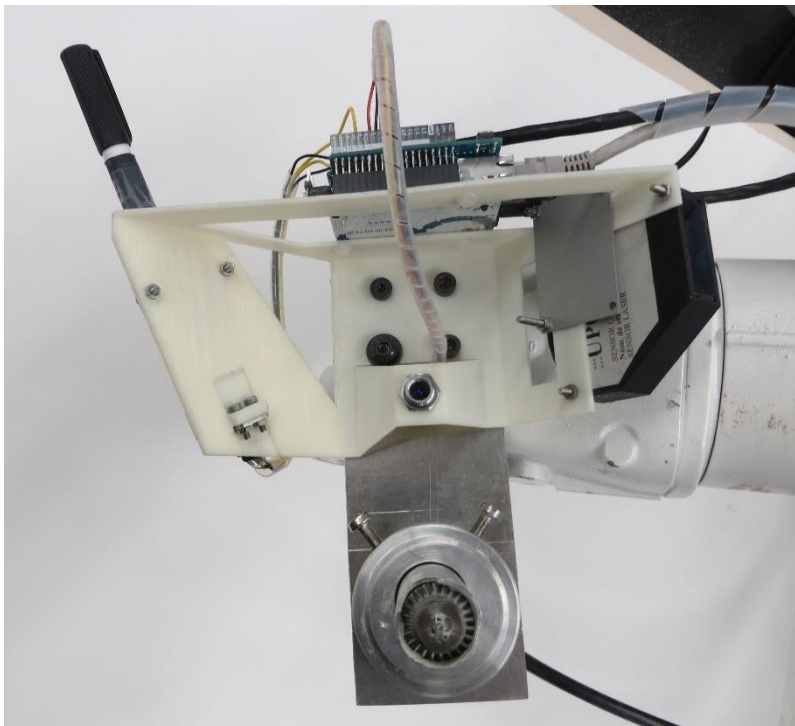
### Disseny i fabricació del capçal

Com ja s'ha exposat anteriorment, per a dur a terme totes les funcions, el robot usa un conjunt d'eines. Per no haver de canviar l'element terminal cada cop que es canviï de funció, s'ha dissenyat un capçal capaç de subjectar-los tots alhora, sense que això suposi una limitació massa important en el rang de moviments del robot.

El capçal està compost per dues peces de materials diferents. Una d'elles és una platina d'alumini que subjecta la polidora i està fixada directament a l'element terminal del robot. L'altra és una peça de plàstic fabricada amb impressió 3D que subjecta l'escàner, el mòdul làser, el marcador i la placa Arduino, i que està fixada sobre la platina.

Per aquesta part del capçal s'ha optat per la impressió 3D ja que aquest mètode permet fabricar amb qualitat un objecte de gran complexitat geomètrica, sense que això suposi un gran cost.

Tot i així el material amb què es fabrica en 3D no és prou rígid per subjectar la polidora. És per això que s'ha optat per afegir la placa d'alumini.



**Figura 3.1:** Imatge del capçal muntat a l'element terminal del robot amb totes les eines incorporades



### 3.1. Platina

La platina s'ha fabricat partint d'una placa d'alumini rectangular. Amb un torn se li han fet els forats necessaris per poder collar-la al terminal i a l'altra part del capçal i subjectar-hi la polidora.

Els forats de subjecció de la polidora són un de cilíndric, per on s'introdueix la Dremel, i quatre circulars, més petits, on es collen els cargols, de mètrica 5.

Els forats encarregats de subjectar la platina amb el robot i amb la peça de plàstic són els mateixos, ja que la placa es col·loca entre els dos. Són quatre orificis circulars, col·locats en forma de quadre, per on han de cabre cargols de mètrica 5. La proximitat del terminal i la polidora pot limitar el moviment, i una distància massa gran entre ells pot dificultar el treball de polit. És per això que per aquesta unió s'han perforat quatre parelles de dos forats per poder collar la platina a l'element terminal en 3 posicions diferents, i així ajustar la separació segons més convingui.

### 3.2. Capçal de plàstic

El disseny del capçal s'ha fet íntegrament amb el programa SolidWorks. Les pautes que s'han seguit en el disseny són:

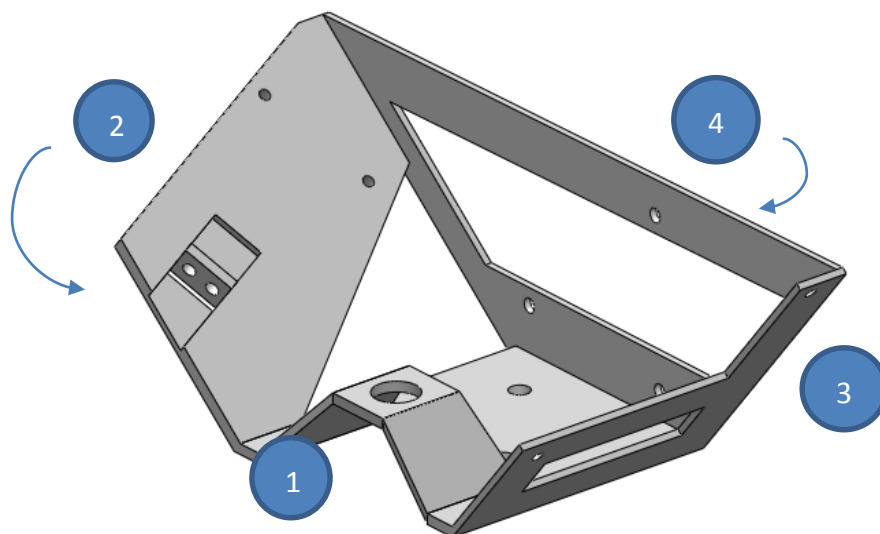
1. **Procurar que les eines no es facin nosa les unes a les altres.** Col·locar totes les eines en el mateix capçal pot provocar conflictes d'espai. S'ha procurat que quan el robot estigui utilitzant una eina, les altres no col·lisionin amb el robot o amb qualsevol objecte que es trobi dins el rang de treball.
2. **Bona orientació de les eines.** S'ha intentat que les orientacions de les eines no forcin el robot a posicions forçades que limitin massa el rang de moviments del robot.
3. **Facilitar el muntatge i desmuntatge del capçal, les eines i el cablejat.** Des que es va muntar el capçal al robot per primer cop, s'ha hagut de desmuntar i muntar varis cops per fer modificacions i correccions. En previsió d'això en el disseny s'ha fet que es pugui desmuntar qualsevol eina sense haver de desmuntar-ne cap altre.
4. **Minimitzar el volum material usat en la fabricació.** Per últim s'ha intentat reduir el volum del material tot el possible sense que afecti negativament a cap

dels punts esmentats anteriorment. El preu de la fabricació de la peça és proporcional al material necessari per fabricar-la, per això reduir-ne el volum n'abarateix el preu.

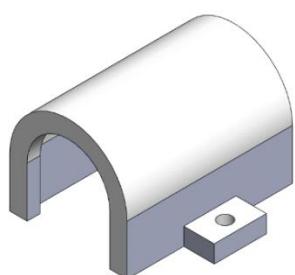
També s'ha fabricat una segona peça que forma part del mecanisme del marcador. La peça té la funció de subjectar el cilindre per on llisca el marcador al capçal.

El disseny final és una peça de  $61,48 \text{ cm}^3$  de volum,  $55,42 \text{ cm}^3$  de la peça 1 i  $6,06 \text{ cm}^3$  de la peça 2. Totes les eines tenen orientacions diferents. Com es pot veure a la figura X, el làser (1), l'escàner (3), el marcador (2) i l'Arduino (4) estan col·locats en quatre cares diferents.

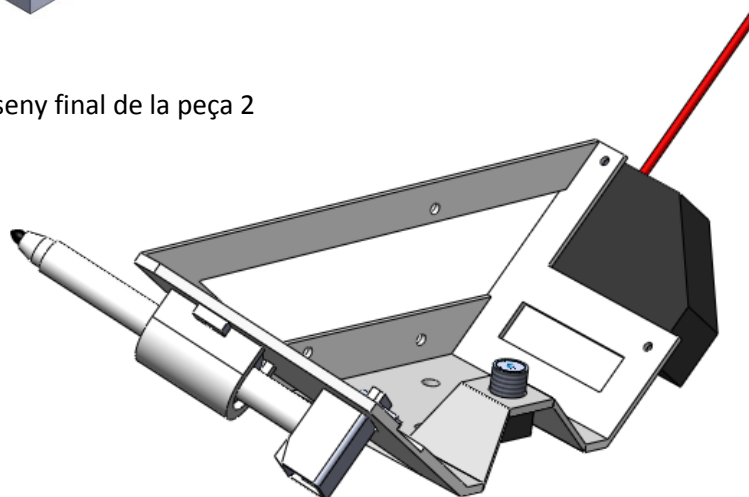
El làser apunta cap a la direcció normal al pla de subjecció amb l'element terminal. L'escàner i el marcador estan orientats en dos plans inclinats per evitar que s'entorpeixin entre ells. El marcador és tan llarg que s'ha hagut d'inclinar respecte un segon eix, ja que si no la punta hauria pogut molestar en l'escaneig i el polit. La placa electrònica està situada a un pla que fa d'unió entre els altres tres.



**Figura 3.2:** Disseny final de la peça 1



**Figura 3.3:** Disseny final de la peça 2



**Figura 3.4:** Muntatge de les dues peces amb les tres eines

Tot i que s'havia marcat com un objectiu, no s'ha pogut evitar limitar el rang de moviments del robot. Algunes posicions que el robot podia assolir, ja no són possibles perquè la part posterior de la Dremel xoca amb el robot. Aquestes posicions depenen únicament de l'angle de gir de les articulacions 5 i 6 del robot, com es pot veure a la Taula 3.1.

Angle Joint 5	Angle Joint 6
$45^{\circ} > J(5) > 90^{\circ}$	$-400^{\circ} < J(6) < -220^{\circ}$ $-140^{\circ} < J(6) < 140^{\circ}$ $220^{\circ} < J(6) < 400^{\circ}$
$-45^{\circ} < J(5) < 45^{\circ}$	$-400^{\circ} < J(6) < 400^{\circ}$
$-90^{\circ} < J(5) < -45^{\circ}$	$-320^{\circ} < J(6) < -40^{\circ}$ $40^{\circ} < J(6) < 320^{\circ}$

**Taula 3.1:** Combinacions dels angles de gir possibles de les articulacions 5 i 6

Per tal que no hi hagi cap col·lisió durant el funcionament de cap dels processos, s'ha afegit en totes les aplicacions una funció que, abans d'enviar el missatge amb la posició al robot, comprova que no es dona cap de les combinacions que resulten en el xoc.



## Capítol 4

### Calibració

La bona calibració de totes les eines utilitzades és bàsica per aconseguir uns resultats de bona qualitat. Com a calibració d'una eina s'entén la seva posició i la rotació respecte el monyó del robot. Una calibració errònia o poc exacte pot no afectar en alguns casos, però pot ser determinant en els casos en que l'error s'acumula al llarg el procés. En aquest projecte s'han de calibrar les quatre eines que es fan servir: l'escàner, la broca, el marcador i el làser. Per a totes elles s'ha fet servir el mateix mètode, amb algunes variacions segons el cas.

El mètode consisteix en l'obtenció d'unes equacions que s'aconsegueixen posant l'element terminal en contacte amb un mateix punt amb diferents posicions del robot. Per cada posició del robot s'obtenen tres equacions, una per cada dimensió de l'espai. Aquestes equacions representen la posició del punt respecte la base del robot. Amb el nombre suficient d'equacions es construeix un sistema d'equacions que permet trobar les variables desitjades. És convenient que el sistema estigui sobredimensionat, ja que el mètode no és exacte, i com més equacions es facin servir, més es minimitza l'error. El nombre d'equacions a obtenir també depèn de cada cas, ja que per algunes cal resoldre 6 incògnites i per d'altres 9.

En els casos de l'escàner i el làser, no es poden ni es necessiten saber tots els elements de la matriu de rotació. L'únic vector que importa és el Z, que defineix la direcció del làser en ambdós casos. Per tant, les incògnites que es volen conèixer són les coordenades cartesianes i els tres elements que conformen el vector z.

En els casos del marcador i la broca aquest sistema només permet trobar la posició relativa. L'orientació es troba amb un sistema que s'explica més endavant.

A continuació s'explica el cas particular de la calibració de l'escàner.

En aquest cas no és possible posar en contacte l'extrem de l'element amb el punt. Per això el que es fa és apuntar amb el làser cap al punt, i usar la distància que llegeix el mateix escàner en el sistema d'equacions. Els components que formen el sistema per a cada posició diferent del robot són:

#### Mesures:

Matriu de rotació i translació de l'element terminal: 
$$\left( \begin{array}{ccc|c} Rm_{11} & Rm_{12} & Rm_{13} & Tm_x \\ Rm_{21} & Rm_{22} & Rm_{23} & Tm_y \\ Rm_{31} & Rm_{32} & Rm_{33} & Tm_z \\ 0 & 0 & 0 & 1 \end{array} \right)$$

Distància marcada per l'escàner:  $\lambda$

#### Incògnites:

Coordenades del punt: 
$$\begin{pmatrix} Tp_x \\ Tp_y \\ Tp_z \end{pmatrix}$$

Matriu de rotació i translació de l'eina: 
$$\left( \begin{array}{ccc|c} 0 & 0 & Re_{13} & Te_x \\ 0 & 0 & Re_{23} & Te_y \\ 0 & 0 & Re_{33} & Te_z \\ 0 & 0 & 0 & 1 \end{array} \right)$$

Els elements incògnita són la tercera columna de la matriu de rotació, la translació de l'eina i les coordenades del punt. Els components x i y de la mateixa no tenen rellevància ja que no influeixen en els càlculs i, com ja s'ha explicat anteriorment, no és necessari trobar-los.

La construcció de les equacions té la forma de la Figura 4.1.

Translació robot - terminal

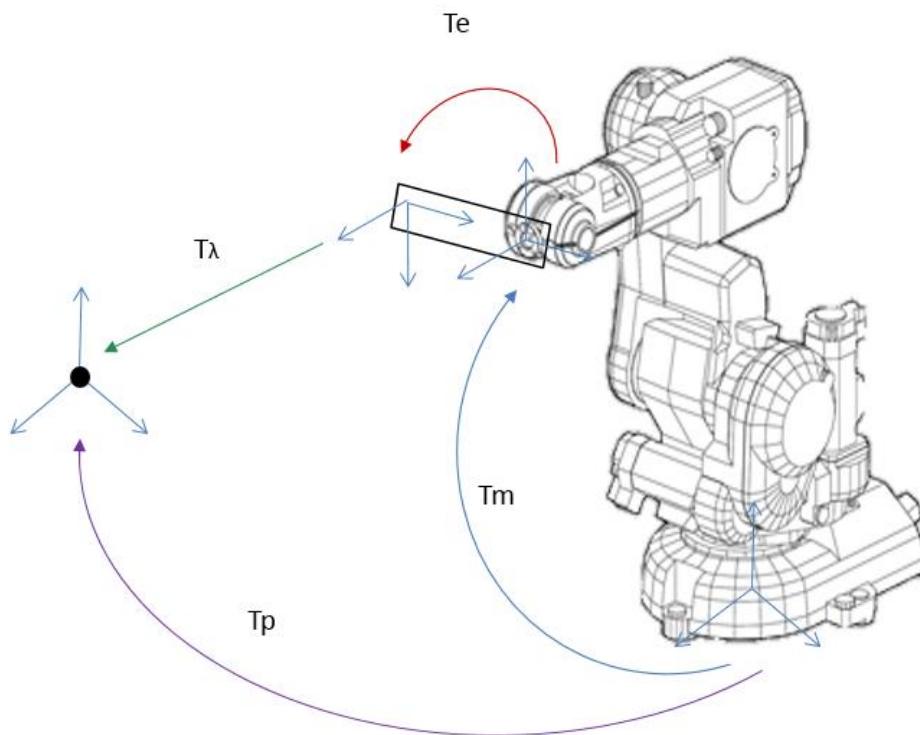
$T_m$

Translació monyó - escàner

$T_e$

$$\begin{aligned}
 & \underbrace{\begin{pmatrix} Tm_x \\ Tm_y \\ Tm_z \end{pmatrix}}_{T_m} + \underbrace{\begin{pmatrix} Rm_{11} & Rm_{12} & Rm_{13} \\ Rm_{21} & Rm_{22} & Rm_{23} \\ Rm_{31} & Rm_{32} & Rm_{33} \end{pmatrix} * \begin{pmatrix} Te_x \\ Te_y \\ Te_z \end{pmatrix}}_{T_e} \\
 & + \underbrace{\begin{pmatrix} Rm_{11} & Rm_{12} & Rm_{13} \\ Rm_{21} & Rm_{22} & Rm_{23} \\ Rm_{31} & Rm_{32} & Rm_{33} \end{pmatrix} * \begin{pmatrix} 0 & 0 & Re_{13} \\ 0 & 0 & Re_{23} \\ 0 & 0 & Re_{33} \end{pmatrix} * \begin{pmatrix} 0 \\ 0 \\ \lambda \end{pmatrix}}_{T_\lambda} = \underbrace{\begin{pmatrix} Tp_x \\ Tp_y \\ Tp_z \end{pmatrix}}_{T_m}
 \end{aligned}$$

$T_\lambda$   
Translació escàner – punt
 $T_m$   
Translació robot - punt



**Figura 4.1:** Representació gràfica de l'obtenció de les equacions per la calibració



Després de modificar les equacions, el sistema, per a cada punt, és el de la figura 4.2.

Per cada posició que es calculi, les matrius A i B augmenten en 3 files. En aquest cas

$$A \cdot X = B$$

**Figura 4.2:** Esquema de les equacions del sistema per a cada punt

amb dues posicions n'hi ha prou per trobar totes les variables. Tot i així, com s'ha comentat anteriorment, convé trobar més posicions per així sobredimensionar el sistema i que els valors trobats siguin més precisos.

Per resoldre el sistema d'equacions s'utilitza el MATLAB. S'ha creat una funció que té com a entrada les dades conegudes, i com a sortida els paràmetres de la figura 4.1. La funció ordena les dades importades d'un fitxer creant les matrius A i B tal com es veu a la figura 4.2. Un cop creades resol el sistema sobredimensionat amb l'equació:

$$X = (A * A')^{-1} * B$$

Un cop obtinguts aquests paràmetres només falta determinar les dues primeres columnes de la matriu de rotació. Com que aquestes no tenen cap rellevància, s'escullen seguint els següents criteris. Per l'eix X s'agafa un vector perpendicular a l'obtingut i paral·lel al pla horitzontal. L'eix Y s'obté fent el producte vectorial dels altres dos.

Després d'obtenir la solució, les coordenades del punt on s'ha apuntat deixen de ser una incògnita, i poden ser utilitzades en altres calibracions si es fa servir el mateix punt, cosa que permetrà resoldre el sistema amb menys equacions.

## Capítol 5

### Pressupost i consideracions ambientals

#### 5.1. Pressupost

El pressupost total del projecte és el resultat de sumar els costos d'enginyeria, els costos materials, i el consum energètic.

##### Costos d'enginyeria

El projecte es realitza principalment per un enginyer superior. La part del muntatge i la soldadura de l'electrònica és realitzada per un tècnic electrònic. S'ha considerat un sou per l'enginyer de 25 €/h, i pel tècnic de 20 €/h.

La durada del projecte ha estat de 6 mesos. Treballant 22 dies per mes, i una mitjana de 5 hores al dia, el total d'hores treballades és 660 ( 594 hores l'enginyer i 66 hores el tècnic).

La feina es distribueix de la manera següent:

- 65% del temps invertit en el desenvolupament de software amb Visual Studio.
- 15% del temps dedicat al disseny (5%) i muntatge (10%) de l'electrònica.
- 10% del temps dedicat al disseny mecànic de l'element terminal.
- 5% del temps fent proves i recollint dades amb el robot.
- El 5% restant correspon al temps invertit en el MATLAB (4%) i l'Excel (1%).

Concepte	Sou (€/h)	Temps treballat (h)	Sou total (€)
Enginyer	25	594	14.850
Tècnic	20	66	1.320
Cost Total			16.170

**Taula 5.1:** Taula explicativa dels costos d'enginyeria

## Costos materials

Els costos materials inclouen la depreciació dels equips utilitzats (un ordinador, el robot i les eines de soldadura), les llicències del programari utilitzat i l'ús de material no re-aprofitable (plaques electròniques, cablejat, cargols, etc.).

S'ha tingut en compte una amortització de l'ordinador de 900€ en 5 anys, del robot ABB de 50.000€ en 10 anys i del soldador de 300€ en 20 anys. La llicència de MATLAB per un any té un cost de 1.380€, la de SolidWorks de 2.000€, la del Visual Studio de 1.283€ i el Microsoft Office de 288€.

Les hores dedicades a cada equip s'han comptat de la següent manera: l'ús de l'ordinador inclou el total d'hores dedicades al projecte menys les de muntatge de l'electrònica (594h). L'ús del robot inclou les hores dedicades a la seva manipulació (33h). L'ús de MATLAB inclou les hores dedicades a la fase de programació amb MATLAB (26h), el SolidWorks les de disseny del capçal (66h), el Visual Studio les de desenvolupament de software (429h) i l'Office les d'ús de l'Excel (6h). A la Taula X es poden veure els costos materials associats a depreciacions.

Per fer el càlcul del temps d'utilització, s'ha considerat que un any equival a 8.760h (365 dies\*24h/dia).

Concepte	Cost total (€)	Vida útil (anys)	Depreciació (€/any)	Temps d'ús (anys)	Cost (€)
Ordinador	900	5	180	0,07706	138,71
Robot	50.000	10	5.000	0,00377	9,43
Llicència SolidWorks	2.000	1	2.000	0,00753	15,06
Llicència MATLAB	1.380	1	1.380	0,00297	4,10
Llicència Office	288	1	288	0,00007	0,2
Llicència Visual Studio	1.280	1	1.280	0,04897	62,68
Cost Total					230,18

**Taula 5.2:** Taula explicativa dels costos de depreciació del material

En la part de costos materials de compra i ús de material no re-aprofitable s'inclou el material següent:

- Plaques electròniques (Arduino Ethernet i ProtoBoard).
- Fabricació del capçal.
- Cablejat, broques, cargols, microxips i demés material.

Concepte	Cost unitari (€/unitat)	Quantitat (unitats)	Cost (€)
Arduino Ethernet	11,76	1	11,76
Arduino Proto Board	54,91	1	54,91
Capçal	54	1	54
Altres material			40
<b>Cost Total</b>			<b>160,67</b>

**Taula 5.3:** Taula explicativa dels costos de material no re-utilitzable

Els costos materials totals són de 390,85 €.

### Costos de consum energètic

Els costos de consum energètic són bàsicament el consum de l'ordinador, el consum del robot i la il·luminació del laboratori. Se suposa un preu constant de l'electricitat de 0,14 €/kWh (TUR sense discriminació horària, Iberdrola). Tenint en compte que l'ordinador consumeix aproximadament 0,25 kW i s'ha usat 594 hores, que el robot consumeix 0,4 kW i s'ha utilitzat 33 hores, i considerant que la part d'il·luminació del total del laboratori corresponent a la realització del projecte són 100 W, resulten els costos de la Taula X.

Concepte	Preu (€/kWh)	Consum (kW)	Temps d'ús (h)	Cost (€)
Ordinador	0,14	0,25	594	20,79
Robot	0,14	0,4	33	1,85
Il·luminació	0,14	0,1	660	66
<b>Cost Total</b>				<b>88,64</b>

**Taula 5.4:** Taula explicativa dels costos de consum energètic

El cost total del projecte és la suma d'aquests tres costos, i és de 16.649,49€, tal com es pot veure a la Taula 5.5.

Concepte	Cost (€)
Enginyeria	16.170
Material	390,85
Consum energètic	88,64
Total	16.649,49

**Taula 5.5:** Taula de suma de costos

## 5.2. Consideracions ambientals

El projecte consisteix principalment en el desenvolupament de software, els únics residus que es generen és el material que s'arrenca del motlle en el procés de polit. Aquest material ha de ser recollit i processat de la manera apropiada segons el tipus de material que sigui.

Un altre concepte a tenir en compte és l'impacte provocat pels equips utilitzats: l'ordinador i el robot, que s'hauran de reciclar quan acabi la seva vida útil seguint les normatives establertes pel *Real Decreto 208/2005, de 25 de febrero, sobre aparatos eléctricos y electrónicos*.

## Conclusions

En aquets projecte s'ha acomplert l'objectiu d'automatitzar les tres funcions que s'havien plantejat: l'escanejat, el polit i el marcat.

S'ha desenvolupat un software amb Visual C++ que permet generar trajectòries i comunicar-se amb el robot i amb la placa electrònica. S'ha dissenyat i muntat un circuit electrònic que permet a la placa controlar i obtenir informació dels dispositius utilitzats en les funcions. S'ha fet el disseny mecànic de l'element terminal que permet realitzar totes les funcions sense haver de canviar el capçal del robot.

S'ha aconseguit crear una aplicació que permet escanejar automàticament objectes amb una gran varietat de geometries amb bona resolució a partir d'informació bàsica sobre la seva forma i la seva posició.

S'ha obtingut un molt bon resultat en la funció de marcar. L'aplicació permet dibuixar traços sobre superfícies irregulars a partir de punts a l'espai.

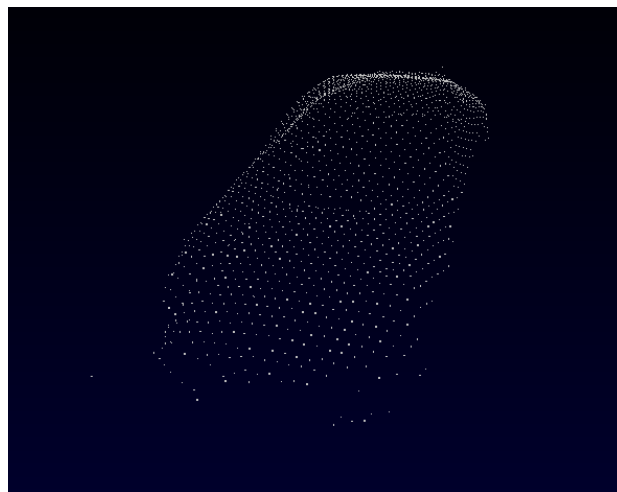
S'ha aconseguit reproduir, mitjançant el polit, la geometria obtinguda en la funció d'escaneig. L'aplicació presenta limitacions en quan a la geometria de l'objecte i la resolució del resultat final. Obtenir una funció sense aquestes limitacions no entrava en l'abast d'aquest projecte i es deixa pendent per un projecte futur.

Pel desenvolupament de software s'ha hagut d'adquirir nous coneixements en programació en C++ amb Visual Studio. També s'ha après a programar i utilitzar una placa electrònica com l'Arduino. S'ha adquirit nous coneixements en el disseny de circuits electrònics i el seu muntatge, que ha requerit l'ús de soldador. També s'ha hagut d'aplicar els coneixements de disseny mecànic per integrar totes els elements en un sol capçal. Aquest capçal s'ha fabricat mitjançant una tecnologia emergent com és la impressió 3D.

Per il·lustrar els resultats aconseguits, a continuació es presenta un exemple del procés escanejat i polit de la Figura 6.1.



**Figura 6.1:** Model de cotxe utilitzat



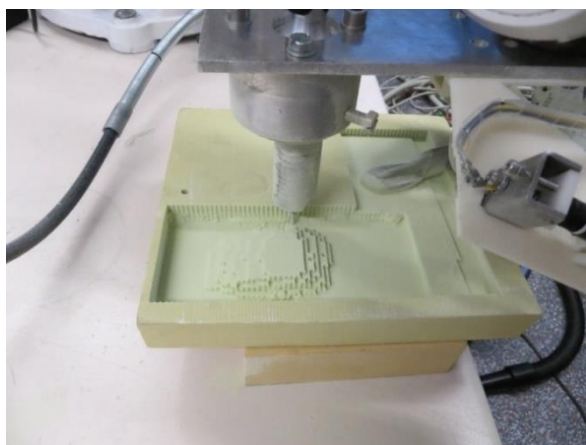
**Figura 6.2:** Núvol de punts obtinguts en l'escaneig abans de ser tractat

La figura és un model de cotxe a petita escala. Les dimensions són 13,5 mm de llargària, 6 mm d'amplada, i 38 mm d'alçada.

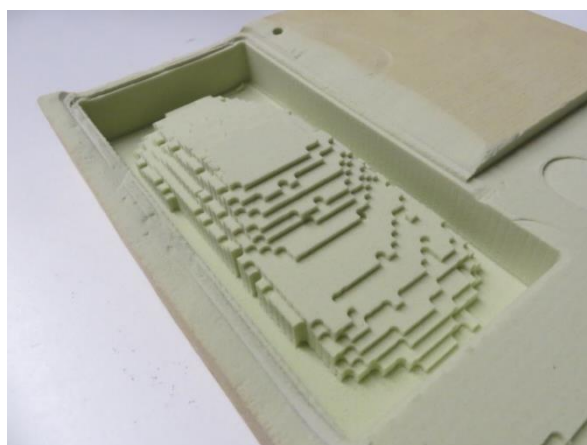
La Figura 6.2 mostra els punts obtinguts en l'escaneig.

Pel procés de polit s'ha fet servir com a motlle una planxa de poliuretà. S'ha indicat la resolució mínima el pla horitzontal, que correspon al diàmetre de la broca (3 mm), i una resolució en l'eix vertical de 2 mm. Aquesta última pot ser tant petita com es desitgi.

La Figura 6.3 és una instantània del procés de polit. En la Figura 6.4 es pot veure el resultat final.



**Figura 6.3:** Moment del procés de polit



**Figura 6.4:** Resultat del procés de polit

## Agraïments

Vull agrair al Xavier Giralt el seguiment constant que ha fet del meu treball i el temps que ha dedicat en formar-me i ajudar-me sempre que m'he trobat en dificultats.

També vull agrair a l'Àlicia Casals haver-me donat la possibilitat de realitzar el projecte al laboratori de robòtica de l'ESAI.

També vull esmentar els meus companys de laboratori, que sempre han estat disposats a ajudar-me quan ho he necessitat.





## Bibliografia

### Pàgines web

<http://new.abb.com/products/robotics/industrial-robots/irb-140>

<http://es.rs-online.com/web/>

<http://www.cplusplus.com>

<http://stackoverflow.com>

<http://www.alldatasheet.com/datasheet-pdf/pdf/73050/MAXIM/MAX233.html>

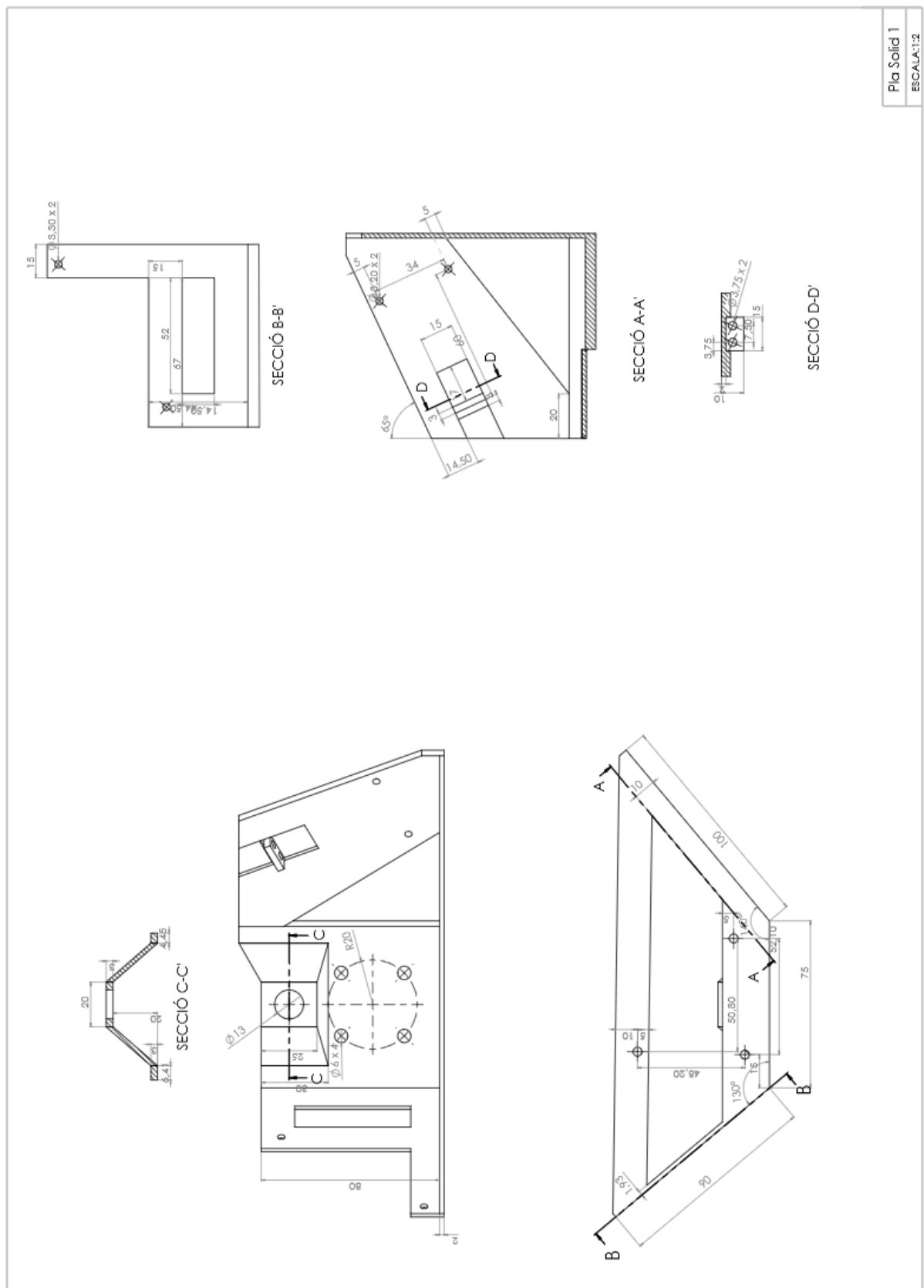
<http://www.arduino.cc/>

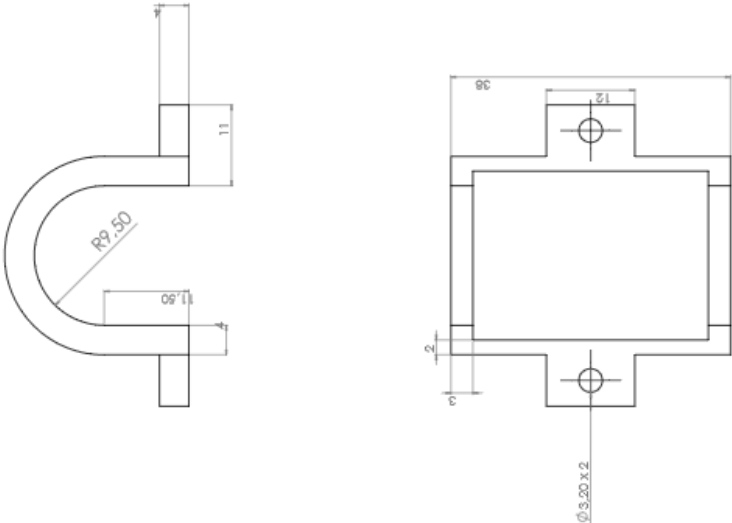
<http://www.3dcontentcentral.com>



## Annex

En l'annex es mostren els plànols realitzats amb el programa de CAD SolidWorks de les dues peces que s'han fabricat amb impressió 3D.





Pla Solid 2  
ESCALA:1:1